Commodore

Cena 12 tys. zł
nr indeksu 355275

1'93

KEFIR

Miesięcznik Użytkowników Komputerów C-64 i Amiga

RAM-dysk (PRG)
CHARBLASTER II
ADPva (23)



GRY NA C-64

AMIGA 1200

KOMPUTEROWY HOROSKOP

Commodore



nr indeksu 355275

Wydawca:
KEBAB - sp. z o.o.
ul. Wojciechowskiego 28
PL - 71-476 Szczecin
telefon (091) 776-74

Redaguje kolegium
w składzie:
Krzysztof Kobus
Patryk Łogiewa
Grzegorz Mikuła
Krzysztof Moroń
Marcin Orłowski
Zbigniew Piotrowicz
Miłosław Smyk
Paweł Sołtysiński

Redaktor naczelny:
Patryk Łogiewa

Szef działu AMIGA:
Krzysztof Kobus
tel. (091) 525-336

Szef działu C-64:
Paweł Sołtysiński
tel. (091) 776-74

Redakcja nie zwraca nie zamówionych materiałów oraz zastrzega sobie prawo wprowadzania zmian w otrzymanych rękopisach.

Wydawca nie odpowiada za treść zamieszczanych ogłoszeń.

Projekt okładki:
Tomasz Kuczyński

Commodore



PRENUMERATA

Każdy egzemplarz zakupiony bezpośrednio u nas kosztuje odpowiednio:
numery: 1; 2/3; 4; 6 - **9,5 tys. zł**
numery: 7/8; 9; 10 oraz następne - **11 tys. zł.**
(**UWAGA!** - nakład numeru 5' 92 jest wyczerpany)

Oznacza to, że można zamówić numery zaległe jak też zaprenumerować jeszcze nie wydane. Odbywa się to tylko poprzez dokonanie odpowiedniej wpłaty na nasze konto.Na odwrocie każdego odcinka kuponu wpłaty należy dokładnie napisać, których numerów wpłata dotyczy.

W przypadku prenumeraty można zamawiać numery do końca aktualnego okresu "małej prenumeraty". W roku 1993 "mała prenumerata" będzie obejmować cztery egzemplarze, z podziałem roku na trzy okresy:

I - numery 1, 2, 3, 4 - 4 egz. x 11 tys. = 44 tys. zł
II - numery 5, 6, 7, 8 - 4 egz. x 11 tys. = 44 tys. zł itd.
(jeśli do tego czasu cena nie ulegnie zmianie).

Prosimy nie przysyłać do redakcji dowodów wpłat.

Nasze konto:
Pomorski Bank Kredytowy
II Oddział w Szczecinie
numer konta: 368113-25771-136

Podajcie dokładny adres, imię i nazwisko zamawiającego oraz numery egzemplarzy, których wpłata dotyczy na odwrocie każdego z odcinków blankietu wpłaty.

Przy okazji prosimy o podanie dokładnego adresu następujące osoby:
RUDNIK ADAM z Warszawy
BAUER (?) ADAM - zam.(?) ul. Targowa 9
GRZEBIELA JANUSZ z Poznania
E. BABULA z Leszna
IGNACIUK ANDRZEJ z Poznania
JARMUSZEWICZ JAN z Gryfic
KUCZKOWSKI STEFAN z Piły

I ostatnia prośba - **wszelką korespondencję, wpłaty etc. kierujcie tylko i wyłącznie na adres redakcji.**

REKLAMA

Ogłoszenia drobne od osób indywidualnych (do 10 słów na kuponie wyciętym z III-ciej strony okładki) przyjmujemy bezpłatnie.
Ogłoszenia drobne od osób prawnych oraz zawierające powyżej 10 słów - 1000 zł za słowo.
Ogłoszenia ramkowe (minimalny format 20 cm²):
1 cm² - 4,5 tys. zł, cała strona 2,5 mln. zł,
cała IV strona okładki - 4 mln. zł. 1/2 tej strony - 2,5 mln. zł,
dodatkowy kolor - odpowiednio 50 % drożej.

Ogłoszenia płatne prosimy przesyłać listem poleconym.



Nr 1 **styczeń 1993**

Supra Fax Modem Plus

- jak z Amigi zrobić telefax ... na stronie10

ADPRO na 1MB?

Proszę bardzo - strona 23

Opis Turbo-Assemblera

znajdziecie na stronie 24

Jak zrobić animację na "REAL'U"

czytaj na stronie 27

Chcesz uniknąć zdziwienia?

- zajrzyj na stronę 12 oraz 15 (Listy do różnych redakcji)

Spis treści :

02 Z kraju i ze świata
03 Amos - część V
04 Ze sceny
05 Assembler na C-64 - przerwania rastrowe
10 Supra Fax Modem Plus i GPFax
12 Odpowiedzi na listy Czytelników (C-64)
13 Co znaczy być w The Silents
15 Listy do różnych redakcji,
16 RAM-DISC-64
17 Nowa generacja
18 Jak zrobić własny turbo-loader, czyli oprogramowanie stacji 1541/1571 część I
23 ADPRO dla "ograniczonych"
24 TURBO ASSEMBLER V5.1
25 Mapa pamięci Amigi - odcinek kolejny
27 REALna animacja
29 DATA TOSTER - "dodatek" do listingu z nru 11-12'92
30 Komputerowy Horoskop
31 Diamenty - gra na C64
32 C-64 GAMES
33 Dr Boczek donosi
Program Niespodzianka
34 Despotyzacja
36 CRAZY BOYS & SKYLIGHT C64 COPY PARTY
37 Listingi:
Amos
Mapa pamięci
Charblaster - w uzupełnieniu do nru 11-12'92
RAMDISC
Dylematy Dr Boczka











NOWOŚCI

NOWOŚCI

W dniach 28-30.12.1992 odbyło się Demo-Party, zorganizowane w Danii przez połączone siły znanych zachodnich grup: Silents-Anarchy-Crystal. Ze względu na aspekt legalności tego typu imprez, odchodzi się już od dotychczasowej nazwy Copy-Party (co z uporem podkreślają sami organizatorzy). Ogrom tej imprezy wyraża choćby liczba obecnych tam osób, która oscylowała wokół 2300(!!!), a także w liczbie prac wystawionych do rozgrywanych tam konkursów. Wyniki tychże konkursów, które nadeszły w ostatniej chwili wprost od jednego z organizatorów, przedstawiamy poniżej.

Do graphics-compo zostały zgłoszone łącznie 53 prace, przy czym każdy z biorących w nim udział artystów, mógł wystawić co najwyżej jedno swoje dzieło. A oto początkowa dziesiątka:

Lp.	Tytuł	Autor/Grupa
1.	WomSnake	Peacher
2.	Encounter	Fairfax
3.	Attack	Groo & Rufferto /Essence
4.	———	Facet
5.	Pamaopa	Rhah
6.	Krabber	Overalt
7.	Splash II	Scuba
8.	Lydie Inspiration	Mack Melon
9.	DayFall (preview)	Domino
10.	Inferno	Nauv

Znacznie większym powodzeniem cieszyło się wśród uczestników music-compo, albowiem do konkursu zgłoszono aż 123 moduły. Ze zrozumiałych względów (123 moduły (średnio) 3 min = 6 godzin (z hakiem)!!!), przeprowadzono wstępną selekcję, tak że ostatecznie można było wysłuchać 34 kompozycje (niestety nie wiemy czy "puszczano" je w całości (co zajęłoby około 1.5 godziny) czy też oszczędzono słuchaczy). A oto jak przedstawiają sie wyniki tej konkurencji:

Lp.	Tytuł	Autor/Grupa
1.	Substance II Tune	Moby/DreamDealers
2.	Hi-Tension	Gurra & Alf / Vitual
3.	Desessart Nicolas	Doh/Delight
4.	Time's the Remedy	Mr.Man/Andromeda
5.	Nicole	Fabian/Desire
6.	Elimination	Lizardking/Alcatraz
7.	Oberheim Power	Emax/TRSI
8.	Technology	Axel/Brainstorm
9.	Riviera	Dean/Cyberiad
10.	Jimi Hendrix	Hein/Vision

W odróżnieniu od schematu znanego z polskich imprez tego typu, na Demo-Party rozegrano także konkurs na najlepsze intro,do którego zgłoszono 25 prac (ciekawostka: żaden z tych programów nie przekracza 40KB długości co pozwoli zmieścić je wszystkie na jednym tylko dysku!).

Lp.	Autor/Grupa	Tytuł
1.	Melon Dezign	Tetris
2.	Shinning	40k Intro
3.	Silents	Intro
4.	Scoopex	Intro
5.	Skandal	Tangentopoli
6.	Rebels	Intro
7.	Shinning/Zeus	Another Revolution
8.	Badcat	Partytro
9.	Electronic Knights	Nearer My God To Thee
10.	Carillon	The Return Of The Son Of The Monster Magnet

Ostatnią kategorią konkursową było czywiście Demo-Compo. Z 24 zgłoszonych prac na czoło wysunęła się nieznana nam bliżej grupa SpaceBalls, wyprzedzając przy tym tak znane grupy jak Anarchy czy Red Sector Inc. Należy przy tym zauważyć pewną, rozprzestrzeniającą się wśród poszczególnych grup tendencję do kontynuacji starszych produkcji. Zaś kolejność przedstawia się nastepująco:

Lp.	Grupa	Tytuł
1.	SpaceBalls	
2.	Anarchy	3D Demo II
3.	TRSI	
4.	Pure Metal Coders	Alpha & Omega Part II "Downtown"
5.	Shining	
6.	Virtual Dreams	Absolute Inebrition Demo
7.	Cult	Kefmania
8.	Paradise Productions	
9.	Lethal Exit	Digital Fr
10.	Future Mirror	Claustrophobia

Oprócz programów startujących w konkursach, na Party pojawiło się także wiele innych programów, które są naszym zdaniem warte odnotowania. Są to: Music-Disk grupy Andromeda, podobnie Solution, gra Public Domain grupy Massive o dźwięcznej nazwie Rectum oraz "preview" nowego PC-Emulatora.

Soeren "[illegible]Sound" Doe[illegible]sing

Jedynym polskim akcentem było, cieszące się dużym powodzeniem, a prezentowane poza konkursem megademo "SOFTWARE" grupy W.F.M.H.

Modelling, Ray-tracing, Morphing, obróbka sygnałów Video to najczęściej pojawiające się ostatnio w doniesieniach światowych hasła. U nas nie inaczej... O ile wydaje się, że wyścig pomiędzy dwoma największymi konkurentami w dziedzinie DTP na Amidze (mam na myśli Pagestream - Soft Logik i ProPage - Gold Disk) jakby trochę zwolnił tempo, o tyle w dziedzinie obróbki i przetwarzania obrazu mamy niesamowity ruch.

ADPro - podstawowy program dla każdego Amigowca mającego ambicje w tym kierunku nie pozostaje już sam na placu boju. Zarówno znany już dobrze ImageMaster, jak i nowy produkt GVP o nazwie Image FX starają się znaleźć swoje miejsce na tym bardzo szybko rozwijającym się rynku. MorphPlus, to uzupełnienie do ADPro wypuszczone niedawno na rynek przez firmę ASDG. Produkt ten z miejsca zdobył sobie ogromne uznanie użytkowników oraz prasy fachowej. Wszyscy, którzy mieli możliwość zetknąć się z efektami jego działania są jednomyślni w pochwałach. Firma ASDG proponuje również demonstracyjną kasetę video, gdzie wszystkie efekty zostały wykonane przy użyciu ADPro i MorphPlus. Ostatnim dowodem na jakość (i szybkość działania) swojego pakietu, jaki przytacza firma jest fakt zakupienia specjalnych stanowisk wyposażonych w Amigi oraz MorphPlus, przez... studia w Hollywood.

Również ImageMaster posiada już kolejną wersję. Wersja ta, oznaczona numerem 9.21 posiada, oprócz możliwości obsługi AA-Chipset'u, znacznie rozbudowany moduł morphingu. Oprócz stosowanych już punktów kontrolnych, można również kontrolować krawędzie transformowanych obiektów, a także przeprowadzać znane z innych programów (MorphPlus, Cinemorph) - Warpingi, czyli wektorowe zniekształcenia obrazu. Innymi istotnymi cechami nowej wersji, jest znaczne zwiększenie szybkości działania (do 10 razy) przy niektórych operacjach, oraz kompatybilność z promowanym przez firmę Soft Logik standardem Hotlinks.

To jeszcze nie koniec doniesień o nowościach w dziedzinie obróbki video na Amidze. Firma bsc (nie pomyłkowo napisana małymi literami) proponuje nową kartę graficzną o nazwie FrameMaster III. Ma ona pozwalać na tworzenie animacji w 24-o bitowej palecie. Jak na razie niestety nie na całym ekranie...

Do karty "FrameMaster III" ma zostać wkrótce zaadaptowane oprogramowanie o nazwie "Onyx", pozwalające na precyzyjną obróbkę obrazów.

Jest już nowy REAL 3D! Zapowiadana wersja 2.0 została gruntownie zmodyfikowana i posiada aktualnie tyle nowych możliwości, że trzeba by było poświęcić cały solidny artykuł tylko na ich wymienienie... Co zresztą wkrótce uczynimy. Na dzisiaj jeszcze zajmiemy się wersją I.4 (na dalszych stronach).

Dla miłośników Imagine'a (i szybkich samochodów), firma CAS proponuje dwa, bardzo szczegółowo wykonane "rodzynki": Ferrari F40 i Lamborghini Countach. Obiekty te są również wyposażone w odpowiednie "Brush'e", które nakłada się na powierzchnie wewnętrzne aby uzyskać efekt oryginalnego wyposażenia wnętrza.

Krytykowany odrobinę przez nas w tym numerze program GPFax posiada już nową wersję. Zgodnie z zapewnieniami producentów, nowa wersja nie posiada już wytykanych w naszym tekście niedociągnięć, a zamiast tego autorzy wprowadzili kilka nowych możliwości. O ile prawdą jest (a dzięki aktualizacji autoryzowanych kopii wkrótce będziemy to wiedzieli), że błędy zostały usunięte, to zanosi się na to, że GPFax wysunie się ponownie na prowadzenie wśród najlepszych programów faksowych o ile nie okaże się, że...

... lepszym rozwiązaniem jest PhonePak oferowany przez firmę GVP. Produkt ten to (zwyczajem GVP) połączony w jednej karcie cały żestaw urządzeń takich jak data-modem, fax-modem, automatyczna sekretarka itp. itd. O ile tylko oprogramowanie dołączane do tego produktu będzie wystarczająco wysokiej jakości, to należy się spodziewać kolejnego przeboju firmy GVP.

Sporo miejsca poświęcaliśmy już w KEBAB'ie rozmaitym emulacjom innych komputerów a przede wszystkim smutnych "pecetów". Spośród software'owych emulacji najlepszą wydaje się być PC-Task autorstwa Chrisa Hamesa, ale również autorzy uznanego pakietu do przenoszenia danych pomiędzy formatami Amigi i MS-DOS'u oraz Atari ST o nazwie "Cross-DOS" postanowili dorzucić swoje słówko w tej dziedzinie. "Cross-PC", tak nazywa się nowy, programowy emulator smutnego świata komputerów pracujących w oparciu o MS-DOS. Cross-PC jest sprzedawany razem z nową (V5) wersją pakietu Cross-DOS. Niestety, jak wynika z niezależnych testów, nie dorównuje on kompatybilnością PC-Task'owi, którego autor pracuje aktualnie nad kolejnymi emulacjami pecetowskich trybów graficznych - EGA i VGA. PC-Task V1.12 na szybkiej Amidze (030 - 50MHz) osiąga Nortonowski CI (Computing Index) równy 3.9 x IBM PC/XT...

Jeżeli komuś za mało jest trzy razy XT, to ma aktualnie do wyboru całą gamę BridgeBoard'ów. Ostatnim hasłem w tej dziedzinie jest Golden Gate 486. Typowy BridgeBoard tyle tylko, że wyposażony w procesor 80486SLC taktowany częstotliwością 25MHz. Producent (Vortex Computersysteme GmbH) wyposaża na życzenie swoje Golden Gate'y w pamięć RAM o pojemności do 16MB, kontroler IDE-AT Bus oraz kontroler do pracy z dyskami elastycznymi poczwórnej gęstości (1.2/1.44MB)

Dotychczas, jeżeli ktoś chciał sobie zafundować wewnątrz Amigi, sprawnego MacIntosh'a, to nie miał specjalnego wyboru. Wysoko oceniany na świecie AMax II+, był jedynym rozsądnym rozwiązaniem. W przypadku Atari ST sprawa wyglądała trochę lepiej. Teraz jednak mamy nowe hasło. EMPLANT. Tak nazywa się nowy, uniwersalny emulator oferowany przez firmę Blue Moon. Uniwersalny, bo dzięki specjalnej, otwartej architekturze sprzętowej umożliwia zainstalowanie na sobie jednocześnie zestawów emulujących Atari ST, Mac'a oraz smutniaka (IBM'a). Oczywiście każdy z zestawów może być również instalowany samodzielnie. Jak TO się sprawuje w praktyce, poinformujemy Czytelników, gdy tylko będziemy mieli możliwość własnoręcznie "pomacać" nowy produkt.

Już wkrótce firma *PSYGNOSIS* ma wydać wersję gry *LEMMINGS* na C64. Nad konwersją pracuje zespół Alter Developments w skład którego wchodza: koder i muzyk z grupy Paradize oraz graficy z grupy Mega Industries. Gra ma ukazać się w sklepach na początku czerwca. Posiadamy w redakcji wersję demonstracyjną (preview) i jesteśmy mile zaskoczeni jakością konwersji.

Grupka studentów ze Stanów Zjednoczonych podjęła się napisania najlepszej i najbardziej rozbudowanej gry role-playing. Samo obmyślenie scenariusza zajęło im 5 i 1/2 roku! Gra ma składać sie z pięciu części z których pierwsza o nazwie MESSIAH III: NEMESIS ukaże się już wkrótce. Tytuł gry jest może trochę mylący gdyż nie ukazały się dotychczas zarówno MESSIAH I jak i II. Pierwsza część będzie zajmować 6 dysków.

W najbliższym czasie firma *Ocean* wyda: *LETHAL WEAPON III*, *WWF* oraz kilka innych gier. Później będzie zainteresowana w wydawaniu na C64 tylko tzw. "dużych" tytułów.

Firma *CODEMASTERS* zapowiada rychłą kontynuację bardzo popularnej serii *DIZZY*.

Amos cz. V.

Stając przed koniecznością napisania programu ilustrującego zastosowanie rozmaitych instrukcji i struktur Amosa, wyszliśmy z założenia, iż tylko taki przykład będzie efektywnie nauczał, który będzie musiał być uzupełniony przez samego Czytelnika. Otóż właśnie mamy zapis programu, do którego należy samodzielnie dopisać dane. Wpisanie i uruchomienie samego listingu da niewiele. Przed "wklepaniem" go do edytora, musimy jeszcze poznać trochę teorii.

Przechowywanie danych w liniach DATA

W Amosie można przechowywać ciągi danych, używając deklaracji Data, np:

```
Data 100, 200, 300, "tekst",
3, 4+5
```

Jak widać możemy przechowywać teksty, liczby i wyrażenia. Ich odczyt przeprowadzamy komendą Read, przyporządkowującą je kolejnym zmiennym. Należy jednak uważać, aby użyte przy Read zmienne były zgodne z rodzajem danych.

```
Read A,B,C,A$

Read D,E

Data 1,2,3,"abc",4,5
```

Po odczytaniu pierwszej wartości z Data, "licznik" danych przesuwa się na kolejną daną. Przy każdym następnym odczycie operacja powtarza się, aż do momentu kiedy nie będzie co odczytać. Gdy "wyczytamy" wszystkie potrzebne nam dane, możemy ów licznik ustawić na początek linii Data komendą Restore, posługując się przy tym etykietą.

Na przykład :

```
For K=1 To 10
For F=1 To 3
Read A
Print A
Next
Restore ETYKIETA
Next K

ETYKIETA: Data 10,4,30
```

Uwaga! Przy używaniu Data w procedurach, dane nie będą dostępne globalnie.

Ponadto do naszego programu potrzebne nam będą jeszcze następujące rozkazy:

Locate X,Y - umieszcza kursor we współrzędnych teksowych X,Y.

Centre "tekst" - działa jak Print, ale centruje tekst (wyświetla go na środku danej linii).

Inkey$ - funkcja przyjmuje wartość aktualnie wciśniętego klawisza.

Proszę teraz zapoznać się z zamieszczonym na końcu numeru listingiem, oraz poniższym komentarzem.

Najpierw trzeba było otworzyć ekran w Hi-Resie (żeby ładniej wyglądało), ściemnić go, schować niepotrzebny wskaźnik myszy, wyłączyć migający kursor i urozmaicić ekran jakimiś ozdobami. Oczywiście nie można było umieścić w listingu grafiki, natomiast procedura OZDOBY stawia dwa kolorowe bary. Jeżeli ktoś chce, może narysować grafikę, i dograć ją zmieniając niniejszą procedurę. Można pokusić się również o otworzenie drugiego ekranu i poszaleć takimi komendami jak Appear, Scroll itd.

Od etykiety STAR: do procedury KONIEC znajduje się główna część programu. Tutaj komputer odczytuje liczbę powtórzeń tego fragmentu, a następnie dane z linii Data, potem drukuje je na ekranie, testuje który z klawiszy od 1 do 5 został wciśnięty i w zależności od tego zmienia wartość zmiennej globalnej WYNIK. Gdy pętla For-Next się skończy, program wykonuje procedurę KONIEC W procedurze koniec podawana jest informacja o wyniku naszej współpracy z programem. Na zakończenie mamy możliwość wyjścia z programu lub jego restartu.

Czy wiecie już po analizie listingu do czego służy ten program? Jeżeli tak to bardzo dobrze, program nie jest skomplikowany ani wydumany, każdy powinien go w pełni zrozumieć. Dalecy jeteśmy od chęci dawania przykładów od których jeszcze bardziej mąci się w głowie. Dla leniwych, którym nie chciało się go przeglądać, oraz dla mniej zainteresowanych podaję, że powyższy program służy do samodzielnego tworzenia testu z dowolnej dziedziny. Po etykiecie DANE: pierwszą wartością w Data jest ilość opisanych przez nas pytań. Następne linie Data uzupełniamy następująco :

Data "Treść pytania",
"odpowiedź1",
"odpowiedź2",
"odpowiedź3",
"odpowiedź4",
"numer odpowiedzi prawidłowej",
"numer odpowiedzi superbłędnej"

Jeżeli przykładowo robimy test z historii, z materiału IV klasy szkoły podstawowej, możemy Data uzupełnić następująco :

Data "Kto to był Kazimierz Wielki?",
"1. Brat Smoka Wawelskiego"

Data "2. Książę z dynastii Ja-

giellonów",
"3. Król Polski."

```
Data "4. Prymas Krakowski",
"3",
"1"
```

Oczywiście nie jest istotne czy wszystkie teksty umieścimy w jednej czy w kilku liniach Data, bowiem ważna jest wyłącznie ich kolejność. Wypełniając linię Data w powyższy sposób, spowodujemy, iż w naszym teście pojawi się zielony napis "Kto to był Kazimierz Wielki?", a pod spodem zostaną wyświetlone cztery odpowiedzi.

Dwie ostatnie liczby (zauważ, że są one zapisane w postaci tekstowej) oznaczają która odpowiedź jest prawidłowa (pierwsza liczba), a która tzw. superbłędna (druga liczba). W naszym przypadku odpowiedzią prawidłową jest odpowiedź trzecia - Kazimierz Wielki był królem Polski, natomiast superbłędna jest pozycja 1 - nawet czwartoklasista powinien wiedzieć, że Smok Wawelski nie miał brata (tylko siostrę - przyp. red.). Za poprawną odpowiedź komputer przyznaje jeden punkt, za błędną zero punktów, a za superbłędną odejmuje karnie 2 punkty.

Jeżeli nie chcemy w danym pytaniu zaznaczać odpowiedzi, za której wskazanie odejmowane będą punkty, to w ostatni nawias wystarczy wpisać zero. Chcąc stworzyć duży test, należy napisać jak najwięcej pytań, policzyć je, wstawić gdzie trzeba ich liczbę i nagrać cały program na dysk. Proste, prawda?

W nagrodę dodatkowo zamieszczamy jeszcze jeden programik. Wielu "amosowców" ma problem z napisaniem scrolla. Poniżej przedstawiamy sposób na w miarę najprostsze rozwiązanie tego problemu. Nowymi elementami w listingu są: funkcja `Space$(x)`, przyjmująca postać ciągu x spacji, oraz funkcja `Len(a$)`, przyjmująca wartość długości wskazanej zmiennej. W tym momencie wyszła na jaw pewna zaległość: nie przerobiliśmy jeszcze komend tekstowych, a były one w tych dwóch programach bardzo potrzebne.

Za miesiąc więc poznamy je wszystkie, oraz znów poanalizujemy gotowe programy. Tymczasem oddajemy do dyspozycji, podobnie jak poprzednio, linie Data. Wpisujemy tam nasz dowolny tekst, a jeżeli dodamy nowe linie, musimy zmienić linię z intrukcją For.

Program 'ten pracuje "w kółko", przerywamy go więc przez Ctrl-c.

To już wszystko w dzisiejszym spotkaniu. W następnum numerze zajmiemy się tekstem, choć oczywiście nie tylko to będzie na "warsztacie". Jeżeli napisaliście jakieś ciekawe programy lub procedury, to przyślijcie je do nas. Najlepsze opublikujemy!

Zbigniew "Zybul" Piotrowicz.

Ze sceny

Również dział Amigi w naszej redakcji doczekał się odpowiedzi na ogłoszony w jednym z poprzednich numerów Kebaba apel "Na scenę!!!". Dzięki temu mam możliwość przedstawić Wam dzisiaj dwie rodzime produkcje. Pierwszą z nich jest gra *STAR FIRE* nadesłana przez grupę **DEATH KISS**. Scenariusz gry przypomina popularną "Space Invaders", gdzie aby przejść do następnego etapu, trzeba zestrzelić wszystko co się rusza i może nas zabić. Po rozprawieniu się z wszystkimi stworko-statko-kosmitami musimy załatwić "mateczkę" i droga do następnego etapu staje otworem. Chociaż gra posiada parę niedociągnięć, z których największym jest długie oczekiwanie na załadowanie kolejnych etapów z dyskietki, to należy podkreślić, że jest pierwszym zupełnie samodzielnym programem napisanym przez tę grupę.



A oto lista osób, które przyczyniły się do stworzenia produktu:

RASTAN - koder;

VADER - grafik;

DEATH KISS nie posiada muzyka, tymczasowo byli więc zmuszeni do "podcięcia" podkładu z innych programów, lecz nie znalazłem informacji z jakich - nieładnie! Jeżeli ktoś zainteresowany jest kontaktem z grupą to może pisać pod adres:

DEATH KISS,

ul. Wielka Skotnica 20/40.

41-400 MYSŁOWICE.

Kolejną nadesłaną produkcją jest demko grupy **Tns** zatytułowane *PSYCHO MEDIUM*. Tak się złożyło, że mam możliwość osobiście już od dłuższego czasu obserwować rozwój tejże grupy, i muszę przyznać, że tym demkiem bardzo mnie zaskoczyli - stanowi ono milowy krok w ich karierze.

W *PSYCHO MEDIUM* możemy oglądać takie efekty jak: pełnoekra-



nowy scroll, bobsy, bobsy połączone z wektorówką, drucianą wektorówkę oraz animowaną kulę złożoną z punktów.

Co prawda w dobie *Mental Hangover, Hardwired*, czy *Software* wspomniane efekty nie rzucają nikogo na kolana, lecz dla wielu koderów w kraju są jeszcze "ideałem niedoścignionym". Wydaje mi się, że chłopaki z TNS powinni popracować jeszcze nad przyspieszeniem swoich procedur, gdyż w niektórych miejscach występują kłopoty z synchronizacją i objekty "skaczą". Grupa zapowiada już demo z wypełnioną wektorówką, ja życzę im wydłużenia czasu ekranowego i podaję autorów oraz kontakt z nimi.

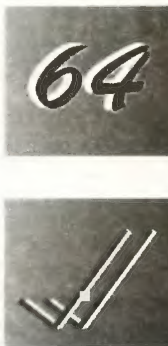
Grafika: **ZEB**

Muzyka: **MR. RAKE**

Kodowanie: **BISHOP, DR SKY**

Bishop/TNS,
ul. Zubrzyckiego 6/40,
44-100 Gliwice.

Krzysztof Kobus



PS.
Jeżeli jesteście autorami demek, gier, użytków i chcielibyście przeczytać o nich na łamach Kebaba to przyślijcie je na adres redakcji. Oczywiście nośniki (kasety, dyskietki) zostaną Wam zwrócone.

Assembler na C-64

- przerwania rastrowe

Z poprzednich odcinków naszego kursu mamy już sporo wiadomości na temat przerwań, oraz kilku specjalizowanych rejestrów VIC'a zajmujących się kontrolą tego co dzieje się na ekranie. Dzisiaj spróbujemy połączyć te wiadomości i zajmiemy się najczęściej stosowanym przez programistów C-64 typem przerwań oraz najczęściej stosowaną techniką kontroli ekranu. Aż dwie sprawy na raz? Nie! Bowiem oba te tematy noszą wspólną nazwę.

Brzmi ona: przerwania rastrowe. Pamiętamy jeszcze z poprzedniego odcinka jak zajmowaliśmy się naszymi pierwszymi przerwaniami? Tam zmienialiśmy tylko wektor pod adresem $0314/$0315. Jednak przerwania te nie miały żadnego związku z obsługą ekranu. Standardowo system ustawia tzw. timer A z układu CIA #1 na generowanie przerwań IRQ co pewien okres czasu. Niemniej jednak istnieje wiele innych możliwych źródeł przerwań. Układ CIA #1 może generować przerwania po odliczeniu określonego okresu czasu - jednorazowo lub ciągle. Inną możliwością jest generowanie przerwania po osiągnięciu określonego czasu przez rejestr TOD (zegar czasu rzeczywistego). Zupełnie innym źródłem może być VIC.

Może on generować IRQ gdy zetkną się ze sobą dwa lub więcej sprite'ów. Również gdy jakiś sprite zetknie się z zawartością ekranu. Można również tak ustawić rejestry VIC'a aby było generowane przerwanie gdy pióro świetlne podłączone do pierwszego portu joystick'a zostanie ustawione w pewnym punkcie ekranu i plamka świetlna tworząca obraz trafi na element światłoczuły. Z naszego punktu widzenia najistotniejszą będzie możliwość generowania, przez układ VIC, przerwania w momencie gdy rejestr RASTER (odpowiadający za pionową pozycję punktu tworzącego obraz) osiągnie określoną przez nas wartość.

Pamiętamy jeszcze naszą pseudo-flagę z jednego z poprzednich odcinków? Tam odczekiwaliśmy aż w/w rejestr osiągnie zadaną przez nas wartość. Robiliśmy to w ten sposób, że porównywaliśmy zawartość akumulatora z zawartością adresu $d012 tak długo, dopóki obie te wartości nie były równe.

Teraz wyobraźmy sobie taką sytuację, że nasz program nie musi się zajmować sprawdzaniem rejestru $d012 tylko ma za zadanie zupełnie inne rzeczy, ale w momencie gdy w rejestrze tym pojawi się żądana wartość układ VIC (dokładniej: VIC-II) poda impuls na wejście IRQ procesora czyli inaczej mówiąc poprosi o przerwanie. W tym momencie procesor zaprzestanie na moment zajmowania się właściwym programem i obsłuży procedurę przerwania.

To wszystko już znamy! Ale jak to zrobić, żeby tak właśnie działały przerwania? Zabieramy się zatem do dzieła!

```
SEI
```

Ta komenda jest nam niezbędna gdyż będziemy modyfikować m.in. wektor przerwań IRQ. Dlaczego, to wiemy już z poprzedniego odcinka. Teraz wiemy, że po każdym resecie (włączeniu zasilania) system ustawia wspomniany już timer A z układu CIA #1 na generowanie przerwań. My zamierzamy przekazać to zadanie układowi VIC. W związku z tym należałoby poinformować układ CIA, że jego usługi nie będą nam w tej chwili potrzebne. Oprócz układu CIA #1 jest możliwe również generowanie przerwań NMI przez układ CIA #2. Na wszelki wypadek poinformujemy i ten układ o zwolnieniu z wszelkich obowiązków dotyczących przerwań.

```
LDA #$7F
STA $DC0D
STA $DD0D
LDA $DC0D
LDA $DD0D
```

Powyższe pięć linijek wykonuje to, o czym przed chwilą wspomina-

liśmy. Należy przy tym zwrócić uwagę, że pokazana wyżej metoda jest zawsze skuteczna. Programiści natomiast często stosują nieco inny sposób wyłączania przerwań systemowych pisząc:

```
LDA #$00
STA $DC0E
```

Tym sposobem najczęściej również uzyskuje się zamierzony efekt jednak nie zawsze musi on zadziałać! Dlaczego? Po prostu w pierwszym przypadku nakazujemy obydwu układom CIA zaprzestania generowania WSZELKICH przerwań. W drugim natomiast przypadku wartość $00 wpisana pod adres $DC0E powoduje tylko zatrzymanie timera odpowiedzialnego za odliczanie okresów pomiędzy kolejnymi przerwaniami systemowymi.

Napisałem, że zazwyczaj to wystarcza, bo najczęściej timer ten jest jedynym źródłem IRQ. Ale może się tak zdarzyć, że jakiś poprzedni program pozostawi nam różne dziwne ustawienia rejestrów układów CIA i okaże się, że nasz program zupełnie nie robi tego, czego od niego oczekujemy. W związku z powyższym zawsze będziemy pamiętać o tych pierwszych pięciu linijkach. Teraz, gdy już mamy wyłączone generowanie przerwań przez układy CIA, musimy poprosić VIC'a aby zajął się tym zadaniem.

```
LDA #$01
STA $D01A
```

Te dwie komendy ustawiają zerowy bit w rejestrze IRQMSK. Pierwsze cztery bity z tego rejestru (0-3) odpowiadają za włączenie generowania przerwań przez układ VIC w określonych przypadkach. Przy czym wpisanie jedynki (ustawienie) do któregoś z tych bitów oznacza włączenie, a zera - wyłączenie generowania przerwań gdy dany warunek został spełniony. I tak:

bit 0 - włączenie przerwań typu "Raster Compare" (to te o które nam chodzi).

bit 1 - włączenie przerwań generowanych gdy nastąpi kolizja (zetknięcie) sprite'a z pozostałą zawartością ekranu.

bit 2 - włączenie przerwań generowanych w momencie kolizji (zetknięcia się) dwóch lub więcej sprite'ów

bit 3 - umożliwia generowanie przerwania w momencie gdy plamka, tworząca obraz na ekranie monitora, trafi na światłoczuły element pióra świetlnego i spowoduje zwarcie linii FIRE w pierwszym porcie joystick'a. Tak na marginesie, to można takie przerwanie wygenerować również przy użyciu joystick'a...

My ustawiliśmy bit zerowy, czyli włączyliśmy tylko przerwania "Raster Compare" zwane w slangu "rastrowymi".

```
LDA #$0F

STA $D019
```

Powyższe dwie linijki również niezbyt często są wykorzystywane przez programistów. Jednak wpisanie ich powoduje, że program nasz będzie ZAWSZE działał poprawnie. Dlaczego? Otóż w momencie spełnienia któregoś ze wspomnianych wyżej warunków, nie następuje bezpośrednio generowanie przerwania lecz najpierw zostaje ustawiony odpowiedni bit w rejestrze VICIRQ pod adresem $D019. Napisałem "odpowiedni", gdyż zależnie od tego, który warunek został spełniony i który bit był wcześniej ustawiony w rejestrze IRQMSK ($D019) tak odpowiadający mu bit z rejestru VICIRQ zmieni swoją wartość na jeden i dopiero ta zmiana któregoś z pierwszych czterech bitów w rejestrze VICIRQ powoduje wygenerowanie przerwania.

Aby to lepiej wyjaśnić, posłużę się drobnym przykładem. Załóżmy, że piszemy grę, gdzie zadanie gracza polegać będzie na poruszaniu się ludzikiem wewnątrz labiryntu i zbieraniu np. truskawek leżących tu i ówdzie na ziemi. W sumie jeden z typowych schematów prostej gry. Nasz ludzik to jeden ze sprite'ów, truskawki stworzymy z pozostałych sprite'ów, natomiast ściany labiryntu to jakaś tam grafika.

Musimy jeszcze dodać, że ściany labiryntu naszpikowane są zatrutymi igłami i jakiekolwiek otarcie się o nie, ma powodować natychmiastową śmierć naszego bohatera. Zatem najprościej: gdy sprite (nasz ludzik) zetknie się z innym sprite'm (truskawką) to truskawka ma zniknąć. Gdy natomiast sprite (ludzik) zatknie się z grafiką (ścianami), to ludzik musi zginąć. Jak zatem sprawdzić czy któryś z tych przypadków zaistniał? Istnieją co prawda specjalne rejestry VIC'a o nazwach SPSPCL ($D01E) oraz SPBGCL ($D01F), w których możemy odczytać czy któryś ze sprite'ów nie zetknął się z jakimś innym lub z pozostałą zawartością ekranu, ale gdybyśmy chcieli skorzystać z nich bezpośrednio, to nasz program musiałby cały czas sprawdzać zawartość tych rejestrów.

Można natomiast posłużyć się przerwaniami. Wtenczas program główny zajmie się np. odczytywaniem pozycji joystick'a i odpowiednim przemieszczaniem naszego ludzika po ekranie, a w momencie gdy nastąpi kolizja ludzika z truskawką lub ze ścianą labiryntu, VIC wygeneruje przerwanie. W tym celu musielibyśmy ustawić bity 1 i 2 w rejestrze IRQMSK. Gdy ludzik zetknie się z truskawką (kolizja dwóch sprite'ów), to w rejestrze VICIRQ zostanie ustawiony bit numer dwa (bo ten typ przerwań włączyliśmy ustawiając bit numer dwa w rejestrze IRQMSK). Gdy natomiast ludzik uderzy o ścianę to w komórce pod adresem $D019 (rejestr VICIRQ) będzie ustawiony bit numer jeden (ten typ przerwań również włączyliśmy wcześniej).

Ustawienie któregoś z w/w bitów spowoduje to, że procesor otrzyma impuls IRQ. Aby jednak można było sprawdzić, który z naszych warunków został spełniony, procedura obsługi przerwań musi odczytać za-

wartość rejestru VICIRQ i sprawdzić który bit spowodował wygenerowanie przerwania. Jak się to robi to w zasadzie już wiemy. Za pomocą komendy AND możemy wykasować wszystkie bity oprócz sprawdzanego i zależnie od tego czy wynik jest równy zero (BEQ), czy nie, podjąć odpowiednie działanie.

Aby np. sprawdzić bit pierwszy, należałoby wykonać co następuje: LDA $D019, AND #$02, BNE... tam gdzie spowodujemy, że nasz ludzik zginie. Jeżeli natomiast okaże się, że bit ten był skasowany, to sprawdzimy jeszcze drugi: LDA $D019, AND #$04, BNE... tam gdzie doliczymy naszemu ludzikowi punkty za zjedzenie kolejnej truskawki. Oczywiście, jeżeli nie wyłączymy wcześniej wszystkich innych możliwych przerwań za pomocą przedstawionych już pięciu komend (LDA #$7F, STA $DC0D... itd.), to może się okazać, że przerwanie zostało wygenerowane nie przez VIC'a lecz przez układ CIA. Również taką ewentualność przewidzieli twórcy C-64.

Wystarczy na początku sprawdzić siódmy (najstarszy) bit rejestru VICIRQ, który przyjmie wartość jeden zawsze, gdy źródłem przerwania jest układ VIC-II. W przeciwnym przypadku musimy szukać winnych za przerwanie pracy procesora wśród rejestrów CIA. No, ale odbiegliśmy nieco od tematu! Wracajmy do naszego podstawowego przykładu. Tak więc, jak już napisałem, dopiero ustawienie (przez samego VIC'a) któregoś z pierwszych czterech bitów rejestru VICIRQ powoduje wygenerowanie właściwego sygnału IRQ. Z kilku bardzo ważnych przyczyn, których nie będziemy tutaj omawiać bit ten tak długo będzie ustawiony dopóki sami go nie skasujemy.

Oznacza to, że jeżeli nasza procedura obsługi danego przerwania wykona co ma do wykonania i zakończy swoją działalność nie potwierdzając tego faktu poprzez skasowanie w specjalny sposób właściwego bitu spod adresu $D019, to natychmiast po powrocie z przerwania wystąpi kolejne, spowodowane tym, że ów bit jest nadal ustawiony. Dlaczego napisałem "w specjalny sposób"? Otóż cały trik polega na tym, że aby skasować któryś z tych czterech bardzo istotnych bitów, nie należy wpisywać do niego wartości zero, lecz... jeden! W literaturze spotyka się takie wytłumaczenie tego fenomenu: Każdy z tych bitów posiada swój własny "zatrzask" (ang.: latch). Gdy zostanie spełniony warunek wywołania przerwania, bit ten jest ustawiany oraz jednocześnie włączany jest jego zatrzask. Dzięki temu zatrzaskowi bit pozostaje ustawiony nawet wtenczas gdy warunek, który spowodował jego ustawienie przestanie być spełniony.

Tak więc wpisując na odpowiednią pozycję (bit) rejestru VICIRQ wartość jeden, nie powodujemy ustawienia tego bitu, lecz jakby zwolnienie odpowiadającego mu zatrzasku, co z kolei pociąga za sobą skasowanie bitu. Teraz zatem nareszcie widzimy co nam daje wpisanie wartości $0F pod adres $D019. Otóż w ten sposób kasujemy wszystkie cztery pierwsze bity rejestru VICIRQ. Dzięki temu jakieś ewentualne pozostałości z przerwań z poprzednich programów nie spowodują nam niepotrzebnych kłopotów.

Następnym krokiem będzie ustalenie w której linii rastrowej (patrz również słowniczki z poprzednich odcinków) ma być generowane przerwanie. W tym celu wykorzystamy rejestr RASTER ($D012), który ma podwójną funkcję. Jeżeli odczytujemy z niego wartości to otrzymujemy informację o tym która linia rastrowa jest aktualnie generowana przez VIC'a. Jeżeli natomiast wpisujemy doń wartości, to ustalamy w ten sposób, w której linii powinno wystąpić przerwanie. Piszmy zatem!

```
LDA #$88

STA $D012
```

W ten sposób poprosiliśmy VIC'a aby był uprzejmy wywołać przerwanie IRQ w momencie, gdy zacznie generować linię rastrową o numerze $88. Pamiętać jednak musimy o tym, że jedna komórka pamięci (bajt), może zawierać wartości tylko od $00 do $FF. Ekran C-64 zawiera natomiast (razem z borderami) więcej linii. Aby można było jednoznacznie odczytać pionową pozycję punktu tworzącego obraz, postanowiono dodać jeden bit więcej. Bit ten mieści się w MSB rejestru SCROLY (pod adresem $D011). Co to jest MSB to wszyscy już mam nadzieję wiemy (jeżeli nie, to należy szybko zajrzeć do poprzednich odcinków naszego cyklu). Jeżeli zatem chcemy być pewni, że przerwanie ma nastąpić w momencie gdy zacznie być tworzona linia o numerze $88, a nie $188, to musimy zawsze pamiętać o tym, że rejestr RASTER jest rejestrem dziewięciobitowym. Oraz o tym, że bit numer osiem tego rejestru mieści się w komórce pod adresem $D011. I jeszcze o tym, żeby nie zapomnieć go skasować (lub ustawić) zawsze gdy wpisujemy coś do naszego rejestru RASTER.

```
LDA $D011

AND #$7F

STA $D011
```



Te trzy komendy rozwiązują nam ów problem. Co jeszcze musimy zrobić, aby uruchomić, w końcu, nasz pierwszy program wykorzystujący przerwania rastrowe? Naturalnie musimy zmienić wektor obsługi przerwań IRQ tak aby wskazywał on naszą procedurę, a nie oryginalną (tą w ROM'ie).

```
LDA #<MOJEIRQ

STA $0314

LDA #>MOJEIRQ

STA $0315
```

Cóż to za dziwolągi? Otóż ktoś spostrzegawczy być może już zauważył, że tym razem nie piszemy przed komendą literki A i adresu. Nie jest to żadna pomyłka! Po prostu przestajemy już powoli pisać nasze programy przy użyciu debuggera i zaczynamy korzystać z "dorosłego" asemblera. W takim przypadku program może działać w różnych obszarach pamięci, zależnie od tego gdzie każemy asemblerowi umieścić nasz kod wynikowy. (Patrz słowniczek)

Zazwyczaj na początku naszego programu musimy umieścić odpowiednią dyrektywę mówiącą gdzie ma się znaleźć nasz program po asemblacji. Najczęściej jest to dyrektywa

```
ORG $????
```

lub

*=$????,

gdzie "????" oznacza czteroznakową liczbę heksadecymalną. Oczywiście do niczego dobrego nie prowadzi umieszczanie kodu wynikowego np. w obszarze rejestrów VIC'a co mogłoby się stać gdybyśmy napisali "ORG $D000" lub "*=$D000" (zależnie od posiadanego asemblera). Na początek proponuje pozostanie przy adresie $5000 do którego już pewnie się przyzwyczailiśmy. Co jednak oznaczają te wspomniane już dziwaczne znaczki. Otóż wyrażenie:

LDA #<etykieta

oznacza załadowanie akumulatora młodszym bajtem adresu pod jakim znajdzie się "etykieta" po asemblacji programu. Pamiętamy jeszcze z poprzedniego odcinka? Tam też wstawialiśmy młodszy bajt adresu naszej procedury obsługi IRQ pod adres $0314. Tyle tylko, że tam musieliśmy wiedzieć od razu pod jakim adresem będzie nasza procedura leżała. Jeżeli korzystamy z asemblera to nie musimy się nad tym zastanawiać. Asembler sam przeliczy wszystko co trzeba i w kodzie wynikowym znajdzie się już odpowiednia wartość. Oczywiście jeżeli nie posiadamy akurat żadnego asemblera, to dziś możemy jeszcze wyjątkowo posłużyć się debuggerem i po założeniu, że procedura "MOJEIRQ" będzie leżała pod adresem $6000 wpisać odpowiednio "LDA #$00" w miejsce "LDA #<MOJEIRQ" oraz analogicznie "LDA #$60" w miejsce "LDA #>MOJEIRQ". Pozostaje nam jeszcze ustawienie początkowego koloru tła dla naszych dalszych poczynań.

```
LDA #$01

STA $D021
```

Tak! To ma być kolor biały. Teraz jeszcze tylko przyzwolenie na przerwania...

```
CLI
```

... i np. powrót do BASIC'a.

```
RTS
```

Zanim jednak uruchomimy nasz program (po uprzedniej asemblacji w przypadku korzystania z asemblera), musimy jeszcze napisać procedurę "MOJEIRQ". Użytkownicy asemblera piszą w tym miejscu:

```
MOJEIRQ:
```

a ci, którzy, po raz ostatni, używają do tych celów debuggera muszą napisać:

```
A6000
```

Teraz właściwa procedura. Od czego zaczniemy? Proponuję rozpocząć od potwierdzenia wykonania obsługi przerwania tzn. od zwolnienia wspominanego już "zatrzasku" w rejestrze VICIRQ. Najlepiej zrobić to teraz, dopóki jeszcze o tym pamiętamy. Potem może się okazać, że długo będziemy się zastanawiali dlaczego program nie chce działać...

```
LDA #$01

STA $D019
```

Tu znowu drobne wyjaśnienie. W tym miejscu znowu różni programiści stosują różne metody. Spotyka się np. LDA $D019, STA $D019 lub nawet INC $D019. W sumie każda metoda, która działa jest dobra. W naszym przypadku jedynymi możliwymi przerwaniami, będą przerwania rastrowe. W związku z tym nie ma specjalnego znaczenia którą metodę zastosujemy.

Osobiście polecam jednak kasowanie tylko tego bitu, który odpowiada za dane przerwanie, gdyż może to się okazać przydatne kiedy będziemy mieli więcej możliwych źródeł przerwań. Ot choćby kolizje sprite'ów z naszej przykładowej gry. Wtencczas będziemy musieli potwierdzać wykonanie obsługi dopiero po rozpoznaniu jakie przerwanie miało miejsce. W naszym przykładzie wiemy od razu, że jest to przerwanie rastrowe, bo tylko takim pozwoliliśmy występować. Wracajmy zatem do przykładu. Przerwanie nasze ma w linii o numerze $88 zmieniać biały kolor tła na czerwony.

```
LDA $D021
EOR #$03
STA $D021
```

Już słyszę jak niektórzy mniej uważni czytelnicy krzyczą: "I to ma być zmiana koloru na czerwony??!" Zgadza się! Pamiętamy o tym, że niedawno wpisaliśmy do rejestru BGCOL0 ($D021) wartość $01? Chyba tak! A pamiętamy z poprzednich odcinków, że przy użyciu funkcji EOR możemy dokonać zamiany stanu poszczególnych bitów? Przynajmniej powinniśmy! Tak więc ładujemy akumulator zawartością rejestru $D021. W rezultacie mamy w akumulatorze następującą wartość:

```
%XXXX0001
```

gdzie bity oznaczone znakiem "X" nie mają znaczenia. Teraz działamy funkcją EOR i zmieniamy stan bitów wskazanych przez argument tej funkcji.

```
%00000011
```

Gdy zerowy i pierwszy bit zmienią swój stan na przeciwny, to w akumulatorze będziemy mieli:

```
%XXXX0010
```

Po odrzuceniu czterech najstarszych bitów jako nieistotne mamy w akumulatorze wartość $2 a to jest właśnie to o co nam chodziło! Tak uzyskaną wartość wstawiamy ponownie do rejestru BGCOL0 i tym

Słowniczek:

Kod źródłowy - to najczęściej tekst programu napisany w dowolnym języku programowania. Kod ten jest następnie bądź poddawany kompilacji (asemblacji) aby uzyskać -> kod wynikowy, bądź interpretowany przez interpreter danego języka.

Kod wynikowy - to efekt kompilacji lub asemblacji programu. Jest to końcowa postać programu po przetłumaczeniu na język zrozumiały bezpośrednio dla komputera.

samym zmieniamy kolor tła na czerwony. Dobrze! Zmieniliśmy już kolor z białego na czerwony, ale jeżeli chcemy aby część ekranu była biała, a część czerwona, to musimy po wygenerowaniu całego obszaru ekranu ponownie przywrócić mu białą barwę. W tym celu musimy albo odczekać aż VIC wygeneruje drugą część ekranu tak jak to już robiliśmy gdy nie używaliśmy przerwań, albo poprosić VIC'a o ponowne przerwanie w odpowiednim miejscu. My zrobimy to drugie i najpierw zaznaczymy w której linii ma wystąpić drugie przerwanie.

```
NOWEIRQ:
LDA #$00
STA $D012
```

Naturalni ci wszyscy, którzy korzystają w tym momencie z debuggera muszą pominąć występującą etykietę "NOWEIRQ". Tym razem nie musimy ustawiać (ani kasować) najstarszego bitu (tego spod $D011) bo zarówno linia $00, jak i $100 znajdują się poza obszarem ekranu. Nie ma zatem specjalnego znaczenia, w której z nich nastąpi przerwanie. Następnie zastosujemy pewien "nieelegancki" trik, który umożliwi nam obsługę dwóch różnych przerwań jedną i tą samą procedurą.

```
LDA NOWEIRQ+1

EOR #$88

STA NOWEIRQ+1
```

W tym miejscu użytkownicy debuggerów muszą oczywiście wpisać gotowe liczby. Należy zatem spojrzeć pod jakim adresem znajduje się ostatnie "LDA #$00", dodać doń $01 i otrzymaną liczbę wpisać w miejsce "NOWEIRQ+1". Jeżeli wpisywaliśmy od adresu $6000, to powinniśmy otrzymać $600E. Co nam te trzy dziwne linijki dają? Otóż, jak już wspominałem, jest to pewien niezbyt "elegancki" trik. Dlaczego nieelegancki?

Otóż pisanie programów tzw. samomodyfikujących się nie należy do dobrego stylu programowania, bo utrudnia ewentualną późniejszą analizę kodu nawet samym autorom. My jednak dzięki zastosowaniu tego triku nie musimy pisać całej, nowej procedury dla drugiego przerwania.

Już wcześniej użyliśmy funkcji EOR, tak aby za każdym przejściem zmieniał się kolor tła z białego na czerwony i na odwrót. Tak, tak!

Gdybyśmy na początku nie wpisali do rejestru BGCOL0 wartości $01, lecz $02 to pierwsze wykonanie się naszej procedury zmieniłoby kolor tła z czerwonego na biały. Polecam wszystkim sprawdzenie tego. Teraz pobieramy bajt będący argumentem komendy "LDA #" i zmieniamy jego wartość z $00 na $88, lub na odwrót (ze $88 na $00), zależnie od tego jaką w danym momencie posiadał. Wszystko to po to, aby przerwanie, które wykona się w linii rastrowej o numerze $00 mogło ponownie ustawić rejestr RASTER na generowanie kolejnego przerwania w linii o numerze $88.

Dokładniejsza analiza kodu powinna wyjaśnić wątpliwości. Co nam jeszcze pozostało? Musimy jakoś wyjść (powrócić) z przerwania. Ostatnio robiliśmy to skacząc pod adres standardowej obsługi przerwań ($EA31). Tym razem możemy zrobić podobnie (a nawet musimy, jeżeli chcemy aby nadal działał BASIC, do którego wracamy komendą RTS z programu inicjującego nasze przerwania).

Nie możemy jednak zapomnieć o tym, że nasza procedura będzie się wykonywała dwa razy w jednym rastrze (patrz słowniczki). To znaczy, że musimy tak zaprogramować wyjście z naszej procedury aby skok pod $EA31 odbywał się tylko raz w rastrze. W przeciwnym wypadku doprowadzimy do efektów nie zawsze pożądanych (polecam sprawdzić). Najprościej w tym przypadku posłużyć się, już wykorzystywaną komórką "NOWEIRQ+1". Dla użytkowników debuggera: $600E.

```
LDA NOWEIRQ+1
BEQ IRQ2
JMP $EA31
IRQ2:
JMP $EA81
```

Jeżeli w komórce tej nie znajduje się wartość $00 to skaczemy do standardowej procedury obsługi przerwań. Jeżeli natomiast jest tam zero, to skaczemy tylko do końcówki tej procedury gdzie pobierane są ze stosu wartości rejestrów A, X, Y oraz następuje powrót z przerwania (RTI). Zainteresowanym polecam sprawdzić debuggerem zawartość ROM'u od $EA81. Tak! To właściwie wszystko!

Pozostaje nam już tylko uruchomienie naszego programu (po uprzedniej asemblacji, jeżeli korzystaliśmy z asemblera). Mam nadzieję, że po lekturze dziesięciu odcinków naszego kursu, każdy z Czytelników wie jak się do tego zabrać...

SD!



Supra Fax Modem Plus i GPFax

Zanim przejdziemy do konkretnego testu, musimy sobie wyjaśnić kilka podstawowych spraw. Modem to ogólnie przyjęty skrót od angielskiego wyrażenia "MOdulator & DEModulator". Modulator, w tym przypadku, to urządzenie posiadające możliwość kodowania danych w postaci impulsów elektrycznych o częstotliwościach akustycznych.

Dzięki zachowaniu wąskiego zakresu częstotliwości akustycznych jesteśmy w stanie impulsy takie przesłać przy użyciu normalnej linii telefonicznej. Podobny sposób kodowania informacji występuje w przypadku przesyłania danych do zapisu na magnetofon (np. Datasette w C-64). Aby tak przesłane dane można było odebrać, potrzebny jest demodulator. Demodulator odbiera sygnały akustyczne z linii, dekoduje je i przesyła gotowe dane do komputera. Tak w największym uproszczeniu wygląda komputerowa komunikacja. Aby mogła ona działać w miarę poprawnie i wiarygodnie, potrzebny jest odpowiedni protokół transmisji.

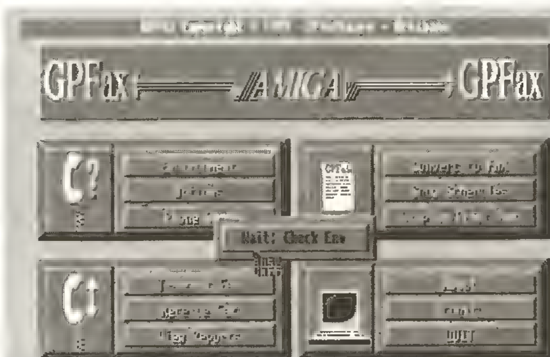
Co to takiego? Jest to pewien zestaw reguł obowiązujących obie strony (nadawczą i odbiorczą) podczas transmisji danych. W zamierzchłych czasach komputerowej telekomunikacji (czytaj: kilka do kilkanaście lat temu), do komputera podłączano dosyć skomplikowane urządzenia spełniające funkcje modulacji i demodulacji. Komputer natomiast, zajmował się nie tylko przesyłaniem danych, ale również ich weryfikacją (sprawdzaniem poprawności przesyłu) zachowaniem odpowiednich odstępów czasowych itp. itd. słowem całym protokołem transmisji.

Dzisiejsze modemy to nic więcej niż wspomniany wyżej modulator i demodulator, który jest zazwyczaj zawarty w jednym układzie scalonym.

W obecnie produkowanych modemach znajduje się zazwyczaj mały komputer z własnym procesorem, pamięcią ROM i RAM, urządzeniami peryferyjnymi (np. głośnik umożliwiający podsłuch linii), układy umożliwiające wybieranie numerów telefonicznych bądź to impulsowo, bądź tonem (DTMF) oraz mały układ zawierający właściwy "modem". Po co to wszystko? Otóż chodzi o to, aby odciążyć nasz komputer od całej obsługi transmisji po stronie linii telefonicznej. Dzięki takiemu postawieniu sprawy komputer może się zajmować tylko transmisją danych do modemu, a ten zajmie się całą resztą tj. dobraniem odpowiedniego protokołu, szybkości itp. a następnie właściwym przesłaniem danych. Gdy uporano się już jako tako z problemami technicznymi, nastąpił lawinowy rozwój komputerowej telekomunikacji (KTK). Jak grzyby po deszczu pojawiały się rozmaite "sieci" komputerowe oferujące swoim użytkownikom "cały kraj pod palcami". Jedną z najpopularniejszych takich organizacji jest pierwotnie amerykańska a dzisiaj już światowa sieć *CompuServe*. Ponieważ okazało się, że **TO** może działać, postanowiono znaleźć inne zastosowania dla KTK. Tak pojawiła się idea maszyny faksymilowej (facsimile machine). Każdy (lub prawie każdy) dzisiaj wie co to jest telefaks. Nie każdy jednak zdaje sobie sprawę, że urządzenie nazywane przez nas telefaksem, to niewielki komputer wyposażony w skaner, drukarkę oraz modem. Ot po prostu! Kartka papieru wprowadzana przez nas do telefaksu jest skanowana, następnie zawarty wewnątrz modem przesyła poprzez linię telefoniczną dane ze skaningu do drugiego modemu (w drugim telefaksie), po czym drukarka (też w tym drugim) drukuje nam kopię naszej oryginalnej kartki.

Proste! Ale ktoś mógłby powiedzieć: przecież komputer już mam, drukarkę też, to może wystarczy jeszcze dokupić skaner i modem i będę miał pełnoprawny telefaks?

Odpowiedź: W zasadzie tak, ale... skaner mający możliwość skanowania strony A4 jest dosyć drogi (najczęściej droższy niż cały telefaks). Oprócz tego niezbędne jest jeszcze odpowiednie oprogramowanie dla danego komputera, tak aby mógł on spełniać wszystkie funkcje telefaksu. Oprogramowanie takie też trochę kosztuje, a poza tym nie na każdy komputer zostało napisane. Czy wobec tego należy od razu zrezygnować z pomysłów wykorzystywania domowego komputera jako domowego telefaksu? Otóż okazuje się, że sprawa nie przedstawia się tak czarno jak by mogło się wydawać. Co prawda nasz komputer to nie żadna profesjonalna maszyna (np. IBM PC AT 286-16) do której doładowujemy profesjonalnie szybki system WINDOWS, lecz zwykła zabawkowa Amiga, ale i na ten zabawkowy sprzęt istnieje już kilka pakietów obsługi modemów faksowych. Najbardziej znane to: *AFax*, *GPFax* i *MultiFax*. Druga sprawa to nie zawsze będzie nam potrzebny skaner, bo większość przesyłanych informacji to po prostu teksty wydrukowane np. na papierze firmowym. Pakiety software'owe pozwalają na konwersję dowolnego tekstu do postaci graficznej z pominięciem wydruku na drukarce.



Trzecia sprawa to drukarka, która również nie będzie nam

całkiem niezbędna, bo dokumenty do wysłania będziemy przygotowywać na Amidze bez konieczności ich drukowania, a dokumenty przychodzące możemy sobie spokojnie obejrzeć (przeczytać) na ekranie i również nie musimy ich drukować. Pozostaje nam już tylko modem i odpowiedni program. Tu dochodzimy do sedna sprawy. Stajemy przed problemem wyboru modemu. Aktualnie istnieje wiele modeli modemów faksowych produkowanych przez różne firmy. My do celów testowych wybraliśmy *SupraFAXModem Plus*.

Przede wszystkim dlatego, że jest to najtańszy model z całej serii modemów produkowanych przez *Supra Corporation* oraz jeden z najtańszych na rynku. Inne , droższe modele produkowanę przez Supra corp. posiadają w zasadzie identyczne funkcje faksowe, a różnią się przede wszystkim funkcjami przesyłu danych (DATA). Co zatem dostajemy w pudełku? Oprócz samego urządzenia (modemu) mamy wewnątrz zasilacz sieciowy, kable przyłączeniowe do linii telefonicznej oraz kabel do połączenia modemu z komputerem. Niestety nie zostało dołączone oprogramowanie. Tylko najdroższy model zawiera w zestawie program GPFax do transmisji faksowej. Fakt ten podnosi koszt zestawu o kwotę niezbędną do zakupu odpowiedniego programu. Tak wyekwipowani możemy już podłączać się do linii.

Oczywiście (?!) musimy posiadać wcześniej odpowiednie zezwolenie itp. itd. Zakładając, że wszystko to już mamy podłączamy radośnie nasz modem do linii przerabiając po drodze bądź nasze gniazda telefoniczne na standard amerykański, bądź kable od modemu na standard "polsko-środkowo-wschodnio-europejski". Gdy już to zrobimy i mimo to mamy ciągle możliwość usłyszenia sygnału w słuchawce (tj. linia nie uległa "zawieszeniu"), to możemy przystąpić do testów. Ah, prawda! Jeszcze musimy zainstalować nasz nowo zakupiony program.

Najlepiej zrobić to na twardym dysku. Gdy i to będzie już za nami, to dzwonimy do znajomego z telefaksem i mówimy "Słuchaj Zenek, ja tam zaraz poślę do ciebie faks, powiesz mi co dostałeś ok?" Program GPFax pozwala na dość sporą dozę arystycznej inwencji w projektowaniu standardowej strony. Możemy sobie skonfigurować nagłówek oraz stopkę a także umieścić weń rozmaite obrazki (IFF). Obrazkiem takim może być np. nasz własnoręczny podpis po przeskanowaniu, a także zdjęcia, znaczki itp. Tekst napisany dowolnym ASCII edytorem (np. ED, CED itp.), można przekonwertować do postaci graficznej. Normalnie program wykorzystuje do tego celu własny font o nazwie GPFax.font, ale nic nie stoi na przeszkodzie aby kazać mu korzystać z innego fontu najlepiej tej samej wielkości. W tym momencie sprawa polskich literek przestaje być problemem (naprawdę!).

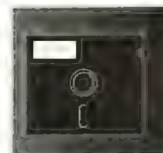
Spróbujmy zatem jak to działa! Przykładową stronę zawierającą zdjęcie, polski tekst oraz podpis (IFF) przesłaliśmy bez kłopotów. Okazało się tylko, że zdjęcie dotarło do odbiorcy w negatywie oraz, że jest nadmiernie wydłużone. To samo stało się z podpisem. Drobny retusz polegający na likwidacji interlace'u oraz zamianie tła i treści obrazka załatwił sprawę, lecz pozostawił nieprzyjemne uczucie niedopracowania programu w szczegółach. Uczucie to pogłębiło się jeszcze przy dokładniejszym poznaniu programu.

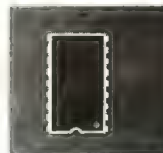
Co prawda zawiera on bardzo dużo funkcji czyniących z naszej Amigi bogato wyposażony telefaks, ale próby skorzystania z przewidzianych przez autorów możliwości konfiguracji programu kończą się często dość nieoczekiwanymi efektami. Przykładowo skorzystanie z proporcjonalnego fontu większych rozmiarów do konwersji tekstów, kończy się... zawężeniem szerokości strony. Literki są większe, ale jest ich zaledwie kilka w wierszu... Od strony hardware'owej modem spisywał się bez zastrzeżeń. Jakość przesyłanych obrazów jest dużo wyższa niż w przypadku normalnego telefaksu. Tak zresztą powinno być! Omijamy bowiem, najbardziej psujący sprawę, proces drukowania na kartce papieru, w Polsce niestety najczęściej przy użyciu drukarek marki "STAR", a następnie skanowania zadrukowanej kartki skanerem zawartym wewnątrz większości popularnych telefaksów.





W żaden niestety sposób nie uda nam się pominąć ostatniego kiepskiego ogniwa w łańcuchu transmisji telefaksowej. Mam tu na myśli jakość naszych łączy. Ogromna ilość zakłóceń (ang.: static) oraz rozmaite zaniki i zniekształcenia sygnału powodują znane wszystkim użytkownikom telefaksów "efekty specjalne" widoczne po odebraniu strony.





Po sprawdzeniu nadawania postanowiliśmy sprawdzić odbiór. "Zenek! puść tam teraz coś do mnie, dobra?" Po jednym dzwonku (lub więcej, zależnie od ustawienia parametrów), zgłasza się modem i po chwili wymiany pisków zaczyna mrugać lampka sygnalizująca pracę twardego dysku. Zaglądamy z ciekawością co my tam (na twardym) mamy. Po, niestety dość długim, oczekiwaniu na zdekodowanie otrzymanej wiadomości, oczom naszym ukazuje się obrazek z podpisem. Wszystko w porządku!

I tu właściwie należałoby zakończyć nasze wywody na temat modemu faksowego. Możemy to zrobić w zasadzie jednym słowem: DZIA-



ŁA! Aby jednak nie pozostawiać Czytelników w niedosycie lub niepewności, spróbujemy podsumować nasze wrażenia. Kombinacja modemu faksowego z odpowiednim programem nadaje się doskonale do przesyłania faksów. Daje nam duże możliwości i pozwala na pominięcie procesu drukowania-skanowania niezbędnego w przypadku korzystania z "normalnego" telefaksu.

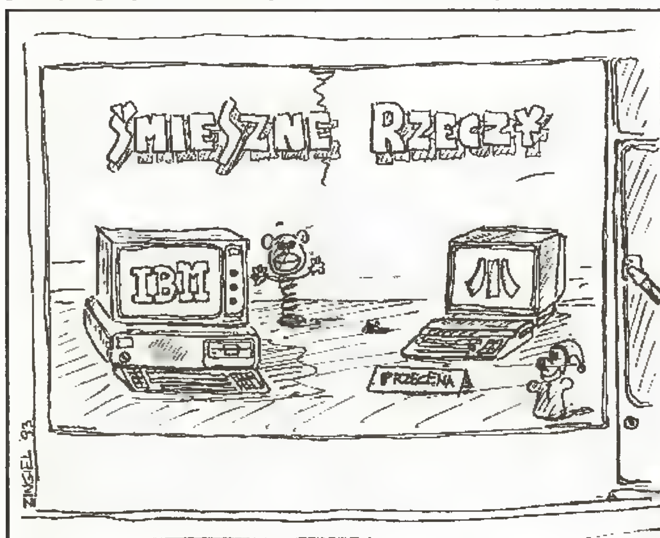
Jedynym problemem staje się przesyłanie gotowych dokumentów. W tym przypadku niezbędnym staje się odpowiedni skaner. Odbiór faksów jest również możliwy i daje dobre rezultaty. Tutaj natomiast problemem pozostaje fakt, że musimy niestety mieć cały czas włączony komputer i uaktywniony program obsługi modemu. Co prawda dla naszej zabawkowej Amigi nie jest to specjalnym kłopotem, bo w tym samym czasie możemy korzystać z kilku (lub kilkunastu) innych programów pod warunkiem posiadania wystarczającej ilości pamięci. Czasami chcielibyśmy jednak wyłączyć komputer lub skorzystać z pamięci zajętej przez program faksowy. Oczywiście wszystkie te kłopoty są do ominięcia. Sprawę zasilania można rozwiązać dokupując odpowiednią przystawkę, która będzie włączała zasilanie komputera po pierwszym dzwonku telefonicznym a modem ustawimy na reakcję np. na piąty dzwonek. Czas pomiędzy pierwszym a piątym dzwonkiem powinien wystarczyć na automatyczny start programu faksowego. Problem pamięci też można rozwiązać dokupując... więcej pamięci...

SD!

Odpowiedzi na listy Czytelników (C64):

Jak zwykle postaram się udzielić tutaj odpowiedzi na najczęściej powtarzające się w Waszych listach pytania dotyczące pracy z Commodore 64. Ponieważ kilkoro z Czytelników prosi o zachowanie anonimowości, zdecydowałem się przyjąć tą zasadę na stałe w naszym kąciku listów. Zadawać pytania i otrzymywać na nie odpowiedzi (o ile są mi one znane) można również telefonicznie (numer mojego telefonu znajduje się w stopce redakcyjnej). Tyle tytułem zwyczajowej informacji - przejdźmy lepiej do pytań:

"Piszę własne programy w kodzie maszynowym. Mój komputer ma zamontowany przycisk resetowania, po wciśnięciu którego na ekranie pojawia się plansza taka jak po włączeniu komputera do sieci. Czy można nad tym zapanować, tzn. czy mógłbym to kontrolować dla własnych celów i np. nie dopuścić do wykasowania komputera?" - Oczywiście, że można. Na odłączenie zasilania nikt nic nie poradzi, dane zapisane w pamięci giną bowiem bezpowrotnie. Na szczęście dla nas istnieje poważna różnica pomiędzy wciśnięciem klawisza RESET a wyłączeniem i ponownym włączeniem zasilania - po wciśnięciu RESET dane w pamięci generalnie nie ulegają zniszczeniu (poza kilkoma bajtami). Można też w prosty sposób przejąć kontrolę nad tym, co robi komputer po wciśnięciu tego klawisza. Wykorzystuje się do tego celu najczęściej pewną, zaplanowaną przez projektantów systemu możliwość. Aby wyjaśnić na czym to polega, musimy zanalizować zasadę pracy cartridge'ów. Jest wiele rodzajów tego typu kart ale wszystkie mają tę samą technikę uruchamiania. Podczas realizowania procedur zwykle wykonywanych po wciśnięciu tego klawisza a prowadzących w efekcie do ukazania się na ekranie planszy "Commodore Basic v2....") komputer sprawdza kilka bajtów począwszy od adresu $8004 i jeżeli ułożą się one w tekst "CBM80" to oddaje sterowanie do procedur zawartych w pamięci cartridge'a. W praktyce w momencie użycia przycisku RESET zawarta we włożonej do EXPANSION PORT karcie pamięć "przysłania" pamięć operacyjną komputera (właśnie od adresu $8000) przez co komputer "widzi" to, co znajduje się w karcie a nie własną RAM. Poprzez takie maskowanie komputer napotyka umieszczony gdzie trzeba tekst "CBM80" (faktycznie znajdujący się w pamięci cartridge'a) i oddaje sterowanie do karty.



Wykorzystując ten mechanizm możemy wpisać sobie owo "CBM80" sami i przejąć obsługę RESET dla własnych celów. Oto sposób wpisania danych do pamięci:

```
:8000 x1 x2 y1 y2 D3 C2
CD 38 30
```

x1 i x2 to młodszy i starszy bajt (wektor) adresu, do którego skoczy procesor po wciśnięciu RESET. Wpisujemy tam adres naszej własnej procedury.

y1 i y2 to z kolei wektor do obsługi przerwania NMI (np. po wciśnięciu kombinacji klawiszy RUN/STOP i RESTORE). Jak widać, nad tym też w ten sposób można zapanować.

Dane D3...30 to bajty tworzące tekst "CBM80". Opisana metoda jest bardzo skuteczna pod warunkiem, że nie używamy aktualnie żadnego cartridge'a; mają one bowiem priorytet w momencie RESET i sterowanie zostaje przejęte zawsze przez nie.

"Jestem posiadaczem edytora do projektowania duszków. Chciałbym wykorzystać je w moim programie edukacyjnym, gdzie potrzebuję dużo zdefiniowanej w ten sposób grafiki. Dane zawierające ich kształty zacząłem zapisywać w pamięci począwszy od adresu 2048 (0800 heksadecymalnie) i zajęły mi one pamięć aż do adresu 4480 (1180 heksadec.). Niestety, ostatnie 6 duszków nie daje się prawidłowo wyświetlić na ekranie (pojawiają się jakieś przypadkowe kształty). Dane zawierające kształty duszków są poprawne. Co zrobić?"

- Muszę przyznać, że jakiś czas temu miałem ten sam problem. Czego ja wtedy nie próbowałem (mało brakowało abym zaniósł komputer do naprawy). Jak się okazało, wyjaśnienie jest dość proste: wszelkie dane graficzne, które powinny być widoczne przez układ wizyjny, a które znajdują się pomiędzy adresami $1000 i $2000, będą maskowane sprzętowo przez umieszczone w pamięci ROM komputera definicje znaków ekranowych. Nie pozostaje wobec tego nic innego poza przeniesiniem danych o kształtach Twoich duszków powyżej adresu $2000. Dotyczy to jednak tylko definicji owych 6 ostatnich duszków, które miały nieszczęście znaleźć się w obszarze od $1000 do $1180, a więc maskowanym przez definicje kształtów liter; reszta (ta od $0800 do $1000) może pozostać na swoim miejscu.

"W instrukcji do Komcia znalazłem między innymi spis kodów sterujących, które użyte jako PRINT CHR$(kod) mogą np. ustawić kolor drukowanego na ekranie tekstu itp. Zauważyłem, że kody 8 i 9 powodują zablokowanie i odblokowanie wciśniętych razem klawiszy C= i Shift (zmiana liter z dużych na małe i odwrotnie). Mój kolega potrafi zrobić to samo nie używając PRINT CHR$ ale nie chce mi powiedzieć jak." - Dla większej wygody użytkownika projektanci Commodore 64 uznali, że większość kodów sterujących powinna być dostępna z klawiatury, co powoduje znaczne przyspieszenie ich wywołania. Prostym przykładem może być zmienianie kolorów (np. wciskając CTRL i 2 zmieniamy kolor kursora na biały). Aby uzyskać w ten sposób interesujące Cię kody użyj kombinacji CTRL+H dla zablokowania i CTRL+I dla odblokowania kombinacji C=+SHIFT.

"Jestem posiadaczem magnetofonu. Niedawno dostałem program przyspieszający wczytywanie gier z taśmy, który nagrywa je w taki sposób, że potem nie potrzebują wgrywania programu typu TURBO do pamięci - wystarczy wpisać LOAD i załadować grę, któ ra po wgraniu sama się uruchomi. Wydaje mi się, system nie jest obojętny dla niektórych gier , gdyż w pewnej ich części po automatycznym uruchomieniu nie można obejrzeć czołowki "hakerskiej" - miga ona przed oczami na krótki moment i uruchamia się gra. Normalnie takie zjawisko nie występuje".

- Myślę, że nie należy winić tu programu przyspieszającego wgrywanie programów z taśmy. Normalnie po uruchomieniu gry ukazywała się czołówka grupy, która dany program odbezpieczyła a przejście do właściwej grupy dokonuje się po wciśnięciu klawisza spacji. Najprawdopodobniejszym wyjaśnieniem jest to, że przypuszczalnie po ukazaniu się nagłówka programu (FOUND nazwa) wciska Pan klawisz spacji, po którym to wciśnięciu ładowanie jest kontynuowane. Otóż owo wciśnięcie jest w buforze klawiatury zapamiętane i najprawdopodobniej po uruchomieniu czołówki jest odczytywane jako wciśnięcie aktualne. Aby tego uniknąć wystarczy po ukazaniu się nagłówka nie wciskać żadnego z klawiszy (ładowanie ruszy "samo" po chwili) lub zamiast spacji nacisnąć klawisz C=, który w tym przypadku pełni podobną funkcję.

"Od pewnego czasu wymieniam się z kolegą programami dla Commodore C64. Dyskietki wysyłamy sobbie pocztą. Koperty specjalnie usztywniamy tekturowymi wkładkami, ale i tak często się zdarza, że są pogięte lub nie chcą się obracać".

- Problem stary jak przesyłanie dyskietek pocztą. Podczas przewozu poczta je pognie (najgorsza wersja wydarzeń) albo zaprasuje (zawarty w kopercie dyskietki nośnik wykazuje duże opory przy próbie obrócenia). Dyskietkę z widocznymi na powierzchni rysami raczej trudno odzyskać i najczęściej udaje się to jedynie fragmentarycznie; przy "zaprasowaniu" dyskietki sprawa jest prostsza. Wystarczy postępować wg wskazówek:

1. Skalpelem lub żyletką rozciąć górną krawędź koperty dyskietki (tą która jest skierowana w naszą stronę w czasie, gdy dyskietka jest włożona do stacji dysków)
2. Delikatnie, chwytająca sam koniuszek suchymi paluchami należy wyciągnąć krążek nośnika na zewnątrz i położyć go w bezpiecznym, czystym miejscu.
3. Pusta kopertę nalęży po prostu rozepchać (dłonią, linijką itp).
4. Włóżyć nośnik z powrotem do koperty i zakleić starannie rozciętą krawędź.

Jak mamy szczęście to powinno to wystarczyć.

"Commodore 64 odkupiłem od kolegi. Niestety ma on jedną wadę: nie "chodzi" mi co druga litera w dwóch rzędach klawiszy. Czy to poważna usterka i jakie mogą być koszta jej usunięcia?"

- Przyczyny takiego zachowania mogą być dwie: uszkodzony tzw. port klawiatury (konieczna wymiana układu scalonego) lub brak prawidłowego kontaktu w łączu klawiatury z płytką elektroniki komputera.



O ile Twój komputer nie jest już na gwarancji to możesz pokusić się o próbę własnoręcznego usunięcia uszkodzenia.

Należy rozkręcić komputer i ostrożnie podnieść do góry samą klawiaturę, uważając by niechcący nie oderwać kabli.

Płytka klawiatury jest połączona z płytą główną za pomocą specjalnej łączki. Często wadliwe, pobrudzone lub naruszone połączenie jest przyczyną opisywanych przez Ciebie kłopotów. Trochę operacji w stylu czyszczenie, prostowanie itp. powinno przywrócić dawną świetność klawiaturze komputera.

Paweł "Polonus" Sołtysiński

Co znaczy być w The Silents

W tym miesiącu, zamiast zwyczajowego opisu jednej z polskich grup - noworoczna niespodzianka. *Progen* z *The Silents Norway* napisał specjalnie dla Was o swojej grupie, o tym jak się do niej dostał i o tym, jak wygląda jego praca.

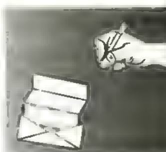
Na wstępie pozwólcie, że się przedstawię. Nazywam się Piotr Ryczko, mam 19 lat i mieszkam w stolicy Norwegii - Oslo. Pierwotnie urodziłem się w Polsce i potrafię płynnie mówić po polsku, czego niestety nie da się powiedzieć o moich umiejętnościach pisarskich (artykuł jest tłumaczeniem z języka angielskiego - przyp.red.).

Moje komputerowe dzieje sięgają czasów gdy będąc trzynasto- czy może czternastolatkiem kupiłem C64 - do gier, naturalnie. Na zabawie minęło mi kilka lat, po czym odkryłem Amigę i natychmiast stałem się jej posiadaczem. Wciąż grałem . Nieco potrwało nim przemogłem się i zainteresowałem się czymś poważniejszym. I dopiero wtedy naprawdę zdałem sobie sprawę z kolosalnych możliwości tej świetnej maszynki jaką jest Amiga!

Zacząłem eksperymentować za pomocą taniego assemblera i jego instrukcji obsługi. Gdy tylko nauczyłem się kodować kilka prostych efektów, wspólnie z paroma zdolnymi chłopakami stworzyliśmy **Massive**.

To już było coś, bawiliśmy się doskonale, lecz mimo to grupa nie miała szans na przetrwanie. Nasz najlepszy grafik - wstąpił do **Static Bytes** w Danii.

Jakiś czas potem zajął się rysowaniem do nowych Eurochartsów i obecnie cieszy się sporym wzięciem (tak na marginesie - pracujemy właśnie nad intrem do gry (nie chodzi tu bynajmniej o "intro" dołączane przez crackerów, lecz o wprowadzenie w akcję jakie podziwialiśmy np. w "Another World" czy "Wolfchild" przyp.tłum.) i naszą własną grą - wszystko dla szwajcarskiej firmy software'owej BasisC\BlackLegend).



Wracając do tematu: cóż **Massive** rozwiało się w nicość. Wróciłem do kodowania na własną rękę, szukając jednocześnie grupy, do której mógłbym (i chciał) się przyłączyć. Bądźmy szczerzy, nawet jeśli natchnienie jest jednym z twoich największych przymiotów, wciąż możesz znudzić się tworzeniem w pojedynkę. Takie przynajmniej jest moje zdanie.

W tamtym czasie otrzymałem zaproszenie na duże party w Norwegii - The Gathering '92, zorganizowane przez grupy **DeadLine** i **Crusaders**. Już wtedy zdecydowałem się na próbę wstąpienia do norweskiej sekcji **Silents**. Na party spotkałem jednego z grafików tej grupy i dobrze się z nim zaprzyjaźniłem. Obejrzał moje procedury i spodobały mu się, pomyślałem więc, że sprawa załatwiona i jestem przyjęty. Nic bardziej mylnego. Powodem był Oystein Eide - jeden z najbardziej uzdolnionych europejskich muzyków, którego chęć przyłączenia się do The Silents i poprowadzenia sekcji norweskiej spełniono w istocie bardzo szybko. To właśnie jego musiałem przekonać, iż jestem wystarczająco dobry. Jak zauważyliście, nigdy wcześniej nie byłem w żadnej znanej grupie i wszystkim, na czym musiałem polegać były me dokonania na Amidze. Sądziłem wtedy, że moje procedury są dobre - zresztą wciąż uważam że takie były - lecz Oysteinowi to było mało.

Dla niego, możesz być naprawdę dobrym koderem, a mimo to niewystarczająco dobrym. Taki już jest perfekcjonista do ostatniego tchu. Jego wymagania? Wysmakowany "design", nowe, wspaniałe pomysły i "kompatybilność" z ludzmi. Na szczęście, chyba znalazł coś z tych rzeczy we mnie, bo po pewnym zastanowieniu pozwolił mi się przyłączyć i wygłosił okazałą mowę witającą mnie w jednej z najlepszych komputerowych grup świata. Wstąpienie do The Silents było naprawdę jednym ze wspanialszych momentów mego życia.

Teraz, będąc w grupie od ponad pół roku, nauczyłem się wiele nie tylko o kodowaniu, lecz także o profesjonalizmie, "design'ie" i innych rzeczach, które uczyniły Silents tym, czym są teraz.

Aktualnie pracuję nad dentrem (prawdopodobnie zwać się będzie **Too Funky**), w którym wykorzystałem kilka bardzo oryginalnych pomysłów. Dentro było oczywiście od samego początku niezwykle dokładnie planowane. Pomysły procedur są głównie moje, choć parę efektów "wymózgowali" także ludzie z **TMB Designs** (twórcy części grafiki do dema **Software** grupy **W.F.M.H.** - przyp. tłum.). Koding jest właściwie na ukończeniu, lecz trzeba jeszcze wszystko wysłać do grafika celem obróbki (kolory, layout, wejście i wyjście). Potem otrzymam nieco uwag na temat pożądanych zmian, wprowadzę je i... koniec.

Pozwólcie, że przybliżę wam tę współpracę na linii grafik-koder. Jeśli dla przykładu zmajstrowałem jakieś zabawne wektory, biorę ripper'a do grafiki, uruchamiam moje procedury i gdy już coś tam namalują, wycinam pamięć ekranu, nagrywam na dysk w formacie IFF i... wysyłam do grafika. Dopiero on dobiera kolory tła oraz obiektów i odsyła mi wszystko wraz ze swoimi ewentualnymi spostrzeżeniami. Proste, co?

Następną sprawą, którą uważam za bardzo istotną jest brak nacisku jeśli chodzi o ILOŚĆ wypuszczanych demosów, intr itd. JAKOŚĆ - oto co się liczy. Nie da się ukryć, że to niezwykle pozytywny i wygodny układ.

Na zakończenie muszę swierdzić, iż jestem bardzo zadowolony i czuję się trochę szczęściarzem, będąc w grupie takiej jak **The Silents Norway**, nie tylko z powodu, że jesteśmy bardzo sławni na scenie, lecz także dlatego, że jestem pomiędzy miłymi i przyjaznymi ludźmi, którzy pomagają mi rozwijać się zarówno towarzysko jak i intelektualnie.

Progen of SILENTS

P.S.: If you are a good coder and would like to write with me for friendship, coding, computers and more then here is my address:

Piotr Ryczko Rustadgrenda 9
0693 OSLO NORWAY

Tłumaczenie:
Miłosław "Thorgal" Smyk



"Listy do różnych redakcji"...

...czyli czego nie powinni byli przeczytać czytelnicy innych czasopism

Pytanie:

Gdzie w Polsce można kupić w miarę tanio dobre dyski twarde do Amigi?

Odpowiedź:

"(...) Osobiście radzę rozejrzeć się za dwoma modelami dysków: spełniającym kryterium niskiej ceny urządzeniem firmy Commodore o nazwie A590(...) (...) lub dyskiem firmy Protar(...)"

Nasza odpowiedź:

Po przeczytaniu powyższych zdań doradzających zakup twardych dysków nasuwa się nieodparcie przekonanie, że autor tych słów miał do tej pory możliwość zobaczenia w działaniu chyba tylko tych dwóch modeli... Być może źródłem takich, a nie innych rad były pewne mechanizmy gospodarki rynkowej...? Poniżej przytoczę krótką listę kompletnych zestawów (kontroler + napęd) z której KAŻDY jest moim zdaniem zdecydowanie bardziej godny polecenia niż wymienione powyżej.

- *Trinology Systems* seria *CHA*. Nadal najszybsze na rynku - z nowymi, wyposażonymi w 256KB bufora napędami firmy Quantum np. LP120, LP240, prędkości transmisji przekraczają 1MB/s. Przelotowa konstrukcja (umożliwiająca np. współpracę z cartridge'ami X-Power i ostatnie modele Action Replay MK-III), wysoka kompatybilność, możliwość zainstalowania wewnątrz obudowy do 4MB RAM, dwóch dodatkowych Kickstart'ów oraz (niestety dość wolnej - 030 16MHz) karty przyspieszającej. Doskonałe oprogramowanie systemowe (m. in. możliwość zabezpieczania hasłem dostępu do całego dysku lub poszczególnych partycji, automatyczne rozpoznawanie wersji systemu operacyjnego umożliwiające start systemu z określonej partycji i wiele innych)

- GVP A500 HD+ (nowa wersja). Wysoka szybkość transmisji, wysoka kompatybilność, możliwość zainstalowania wewnątrz obudowy do 8MB RAM oraz specjalnej wersji emulatora ATonce Plus zwanego GVP PC286, jak również szybkiej (030 40MHz) karty "turbo" również firmowanej znaczkiem "GVP".

- SUPRA 500XP. Bardzo wysoka kompatybilność. (Supra podobnie jak GVP posiada specjalny numer autoryzacyjny Commodore na produkcję peryferiów dla Amigi). Przelotowa konstrukcja, która w połączeniu ze wspomnianą już kompatybilnością zapewnia bezproblemową pracę wszelkich urządzeń dodatkowych dołączanych do Amigi. Możliwość zainstalowania do 8MB RAM wewnątrz obudowy. Rozbudowana obsługa łącza SCSI umożliwiająca np. skonfigurowanie logicznego numeru urządzenia dla kontrolera.

- *Roc Hard*. Niska cena (ca. 7mln z napędem Seagate 80MB AT-bus). Możliwość zainstalowania do 8MB RAM wewnątrz obudowy. Bardzo cicha (mimo zainstalowanego wentylatora) praca.

Wszystkie wymienione zestawy są do nabycia (zazwyczaj na zamówienie) w Polsce i przeznaczone są tylko do A500 lub A500+. Tak zresztą sugerowała nam cytowana odpowiedź. Amiga to jednak nie tylko A500 a użytkownicy np. A2000 mają zdecydowanie większy wybór.

Jednak i tu polecałbym przede wszystkim produkty uznanych firm, posiadających wyżej wspomniane numery autoryzacyjne Commodore. Dobrym przykładem może być *GVP Impact Series II* lub dla bardziej ambitnych *G-Force '030 combo* czyli zintegrowane na jednej sześciowarstwowej (!) płycie rozszerzenie pamięci, kontroler SCSI, napęd Quantum oraz "coś z kopem" czyli 68030 z zegarem do 50MHz. Niezależnie jednak od modelu Amigi radzę trzymać się jak najdalej od oryginalnych produktów Commodore z dziedziny twardych dysków.

Pytanie:

Czym są rozszerzenia fontów do PageStream'a 2.0? Które z nich są konieczne do współpracy z drukarką?

Odpowiedź:

"Dodatkowe fonty, które otrzymuje się wraz z wymienionym programem w odróżnieniu od czcionek systemowych, są zdefiniowane w sposób wektorowy. Oznacza to, że kontur każdej litery jest opisany poprzez odcinki linii prostych, łuków i krzywych(...)"

itp.... itd.... etc....

"(...) Zastosowanie w programach DTP liter konturowych stanowi główną przyczynę ich wolnego działania oraz długiego czasu tworzenia wydruku."

Nasza odpowiedź:

Dodatkowe fonty, które otrzymuje się wraz z wymienionym programem, podobnie jak nowe czcionki systemowe (OS 2.0), są zdefiniowane w sposób wektorowy. Niemniej do rozmaitych celów wykorzystuje się różnie reprezentowane definicje. Oprócz wersji wektorowych znajdują się tam również specjalne wersje bitowe (lub "bitmap'owe") umożliwiające szybszą obsługę ekranu. Zależnie od konfiguracji naszego systemu będziemy korzystali z różnych wersji fontów. Oznacza to, że nie wszystkie są nam jednocześnie potrzebne i część z nich możemy po prostu skasować. Aby jednak nie mieć wątpliwości które, przedstawiamy poniżej poszczególne rozszerzenia nazw i znaczenie zbiorów nimi opisanych.

.DMF - zbiory opisane tym rozszerzeniem to główne definicje wektorowe fontów typowe dla PageStream'a. Są używane do wydruków na drukarkach matrycowych (igłowych, atramentowych) oraz drukarkach laserowych nie wyposażonych w język PostScript lub jego emulację.

.FM - to zbiory zwierające tzw. metrykę (niezbyt szczęśliwe sformułowanie) fontu. Znajdują się tam wymiary poszczególnych liter oraz tzw. tablice Kerningu, jak również odpowiednia nazwa fontu. Te zbiory są niezbędne jeżeli mamy zamiar wogóle korzystać z fontu

.PS - to PostScript'owe definicje fontu. W połączeniu ze zbiorami o rozszerzeniu ".PSF", umożliwiają wydruki na drukarkach wyposażonych w PostScript. Nie są konieczne w przypadku wydruków na pozostałe drukarki.

.PSF - to dodatkowe tablice umożliwiające wydruki przy użyciu języka PostScript. Podobnie jak ".PS" nie są nam potrzebne jeżeli nie używamy drukarki PostScript'owej.

.??H i .??I to wspominane wyżej bitowe reprezentacje fontów umożliwiające szybszą obsługę ekranu. Jakkolwiek nie są niezbędne do pracy, to jednak znacznie ją uprzyjemniają. Zaleca się pozostawienie na dysku roboczym przynajmniej najczęściej używanych wielkości.

Wydaje nam się, że autor pytania oczekiwał mniej więcej tego typu odpowiedzi, chociaż w pewnym sensie mylące może być sformułowanie "rozszerzenia fontów". Naszym zdaniem powinno ono brzmieć "rozszerzenia nazw zbiorów..."

Pytanie:

Co oznacza asemblacja i disasemblacja?

Odpowiedź:

"Asemblacja oznacza zamianę kodu źródłowego programu napisanego w języku wysokiego poziomu (C, Pascal, Basic) na mnemoniki asemblera (nie mylić z kodem maszynowym), disasemblacja jest operacją odwrotną."

Luuuudzieeee! To nie Prima-Aprilis! Takie odpowiedzi ukazują się na łamach czasopism komputerowych w drugiej połowie roku (daleko od kwietnia).

Nasza odpowiedź:

Asemblacja jest procesem zamiany mnemoników języka symbolicznego (asemblera) na wewnętrzny kod maszynowy określonego procesora. Zazwyczaj dokonuje się tego przy pomocy specjalnego programu zwanego również asemblerem. Proces asemblacji można uznać za szczególny rodzaj kompilacji, jednak terminem "kompilacja" określa się przede wszystkim zamianę kodu źródłowego języka wysokiego poziomu (Pascal, C) na język maszynowy procesora.

W żadnym wypadku terminem asemblacja nie można określić zamiany kodu źródłowego programu napisanego na przykład w C na kod maszynowy czy "mnemoniki asemblera". Natomiast termin disasemblacja dotyczy oczywiście procesu przeciwnego do asemblacji, czyli zamiany programu w języku maszynowym procesora na symboliczny język asemblera.

W związku z dwuznacznością słowa "asembler" wynikają czasem duże nieporozumienia. Proponuję więc na określenie języka symbolicznego używanie terminu "język asemblera", natomiast programu do asemblacji po prostu słowa "asembler". Zatem możemy powiedzieć: "Na asemblerze "AsmOne" programujemy w języku asemblera".



Ram-Disk-64

Jak zapewne wielu z Czytelników wie, programy napisane w BASIC V2 na Commodore 64 nie mogą zajmować całej pamięci tego komputera. O ile cała przestrzeń adresowa znajduje się pomiędzy adresami $0000 a $FFFF, to na potrzeby BASIC projektanci systemu przewidzieli obszar od $0800 do $A000. Jak widać, od adresu $A000 do końca pamięci mamy jeszcze $5000 wolnej przestrzeni do wykorzystania; programy pisane w BASIC mogą ten obszar co najwyżej zaadresować przy użyciu instrukcji POKE lub PEEK.

Są programy, które tą górną granicę pamięci dostępnej dla BASIC ($A000) są w stanie przesunąć wyżej (tak to czyni m.in. Warsaw Basic) ale w tym artykule zajmiemy się innym wykorzystaniem tego "leżącego odłogiem" obszaru. Moją propozycją jest tzw. Ram Dysk. Co to jest? Jest to krótki programik, którego zadaniem jest organizowanie owej "dodatkowej" pamięci na zasadzie uproszczonej w działaniu stacji dysków.

Zaletą tego rozwiązania jest szybkie przesyłanie danych (tylko operacje na pamięci z wyłączeniem jakichkolwiek innych urządzeń zewnętrznych) i niebagatelna możliwość użycia owego Ram Dysku jako swojego rodzaju "śmietniczki", gdzie np. można co jakiś czas zapisywać poprzednią wersję pisanego przez nas programu lub parę często wykorzystywanych programów.

Sama procedura leży w pamięci od adresu $9A80 i rezerwuje na swoje potrzeby pozostałą pamięć (aż do jej końca, czyli do adresu $FFFF). Oprócz tego na końcu KEBABA Czytelnicy będą mogli znaleźć tekst źródłowy tego programu spod assemblera, co pozwoli na ewentualne przeniesienie RAM DYSKU pod inny adres. Może to być ciekawym rozwiązaniem zważywszy na fakt, że program alokuje pamięć zaraz "za sobą", co pozwala na zwiększanie bądź zmniejszanie obszaru roboczego.

Uruchomiony program ustawia tzw. pointery interpretera BASIC tak, aby otwieranie zmiennych tekstowych w pamięci nie uszkodziło samego programu obsługi Ram Dysku. Modyfikuje on również systemowe wektory LOAD i SAVE a także procedurę GET-CHARACTER spod adresu $0073. W ten sposób rozkazy LOAD i SAVE odnoszą się do Ram Dysku, pod warunkiem, że numerem urządzenia będzie 12. Np.:

SAVE"nazwa",12 - nagrywa aktualny program do Ram Dysku

LOAD"nazwa",12 - ładuje program z Ram Dysku pod adres początku programu w języku BASIC (wskazany przez wektor $2B/$2C);

LOAD"nazwa",12,1 - ładuje program z Ram Dysku pod oryginalny adres (ten, spod którego program ten był kiedyś odczytany);

Ram Disk 64 dodaje także 3 nowe komendy:

←$ - wyświetla nazwy i długości zbiorów zapisanych w Ram Dysku, wraz z ich długościami i ilością wolnej pamięci;

←C - kasuje zawartość Ram Dysku;

←D"nazwa" - kasuje z Ram Dysku zbiór o nazwie podanej pomiędzy znakami cudzysłowia;

Kilka informacji, które powinny być pomocne przy korzystaniu z Ram Dysku:

- dane zapisywane są w pamięci w postaci skompresowanej, co pozwala na efektywniejsze gospodarowanie przeznaczoną na Ram Dysk pamięcią;

- Ram Disk 64 najprawdopodobniej NIE BĘDZIE prawidłowo pracował z ŻADNYM cartridge'm!

- przy komendach LOAD, SAVE i kasowaniu danych z Ram Dysku obowiązują pełne nazwy programów;

- należy uważać, żeby nieopatrznie nie zniszczyć zawartości pamięci przeznaczonej na Ram Dysk (np. nie uruchamiać Tape Turbo Rom itp.)

- ilość programów zapisanych na raz w Ram Dysku jest ograniczona i wynosi 16

- komenda SYS 39552 ponownie uaktywnia Ram Dysk (np po wykonaniu komendy SYS 64738, czyli po resecie) i kasuje zapisane w nim dane;

- komenda SYS 39552+3 uaktywnia Ram Dysk bez kasowania zapisanych w nim danych;

- Ram Disk 64 przejmuje kontrolę na przerwaniem wywoływanym przez kombinację klawiszy RUN/STOP i RESTORE;

Sam program należy wpisać do pamięci przy pomocy KOREKTORA lub dowolnego debbugera, jednak wtedy należy pomijać (nie wpisywać) umieszczonych na wydruku sum kontrolnych,zapisanych w nawiasach.

Paweł "Polonus" Sołtysiński

NOWA GENERACJA

Pod tym hasłem Commodore reklamuje swój najnowszy produkt - Amigę 1200, który to model ma zastąpić starą, poczciwą pięćsetkę. A1200 jest komputerem domowym, czyli tanim - jej cena w Niemczech nie przekracza w wersji podstawowej 1000 DM, a za odpowiednio wyższą kwotę otrzymać możemy także wbudowany dysk twardy.

CO Z WIERZCHU ?

Nowy komputer prezentuje się z zewnątrz dość przyjemnie - jest mały, niewiele większy od samej klawiatury. A1200 posiada obudowę podobną do A600, lecz układ klawiatury identyczny jest z A500 - posiada wydzielone klawisze kursora i blok numeryczny. W komplecie znajduje się także nowa, bardziej ergonomiczna mysz.

A1200 posiada większość standardowych dla A500 złącz, a także jest na wzór A600 wyposażona w modulator, w związku z tym, na tylnej ściance znajduje się gniazdo RF i kolorowe Composite Video.

Dotychczasowe złącze Expansion z lewej strony obudowy zastąpione zostało przez gniazdo PCMCIA 2.0 (Personal Computer Memory Card International Association). Umożliwia ono podłączenie standardowych kart pamięci wykorzystywanych między innymi w laptop'ach. Mogą one funkcjonować jak szybki dysk (pamięć z podtrzymaniem bateryjnym) lub symulować fast-RAM (będzie on jednak wolny, ponieważ PCMCIA jest standardem 16 bitowym). Maksymalnie możliwe jest podłączenie w ten sposób 4 MB pamięci, także ROM. Z tego gniazda korzystać mają w przyszłości kontrolery SCSI, karty sieciowe, faxowe i inne.

Zamiast złącza rozszerzenia pamięci znajdującego się pod spodem komputera, A1200 ma w tym miejscu wyprowadzoną szynę procesora, która ma umożliwiać podłączenie pamięci fast, koprocesora, bateryjnego zegara, itp.

CO W ŚRODKU ?

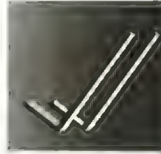
Amiga 1200 jest wyposażona w procesor 68EC020 taktowany częstotliwością 14.2 MHz, wyposażony w 256 bajtów wewnętrznej pamięci cache. Jest on w pełni 32 bitowy, szyna danych ma szerokość 32 bitów, natomiast adresowa - 24. Standardowo A1200 posiada 2 MB pamięci typu chip. Płyta komputera wykonana jest w technologii montażu powierzchniowego, co, jak wiadomo, zwiększa niezawodność, lecz podnosi koszty ewentualnych napraw.

Na płycie znajduje się zintegrowany kontroler dysku twardego IDE (AT-BUS), a także przewidziane jest wewnątrz obudowy miejsce na montaż dysku 2.5 cala. Jego instalacja jest prosta - wystarczy go zamocować w ramce i podłączyć wtyczkę.

W A1200 zainstalowane zostały nowe układy specjalizowane - Alice i Lisa, znane jako AA-chipset - identyczne jak w A4000. Dają one Amidze wspaniałe możliwości graficzne oferując 24 bitową paletę barw. W każdej rozdzielczości mogą one wyświetlić 256 kolorów z palety 16777216, lub 262144 naraz w trybie HAM8. Dostępne rozdzielczości nie uległy zmianie w stosunku do ECS. Dzięki programowalności układów graficznych, A1200 dysponuje między innymi trybami (cyfry po prawej stronie oznaczają częstotliwość ramki i odchylania poziomego):



PAL:Super-High Res Laced 1280x512 50 Hz, 15.63 kHz



NTSC:Super-High Res Laced 1280x400 60 Hz, 15.75kHz

DBLPAL:High Res No Flicker 640x512 48 Hz, 27.50 kHz

DBLPAL:High Res Laced 640x1024 48 Hz, 27.50 kHz



DBLNTSC:High Res No Flicker 640x400 58Hz, 27.66 kHz

DBLNTSC:High Res Laced 640x800 58 Hz, 27.66 kHz

EURO:72Hz Productivity 640x400 69 Hz, 29.32 kHz

MULTISCAN:Productivity 640x480 58 Hz, 29.29 kHz

SUPER72:Super-High Res Laced 800x600 71 Hz, 23.21kHz

Oprócz pierwszych dwóch pozostałe rozdzielczości wymagają monitora typu multiscan lub multisync. Posiadając taki monitor możemy zmusić system operacyjny, aby wszystkie ekrany zamiast w interlace były otwierane np. w trybie DBLPAL, dzięki czemu zbędny jest flicker-fixer.

WORKBENCH

Amiga 1200 wyposażona jest w Kickstart 3.0, który został rozbudowany w stosunku do 2.1 głównie o obsługę nowych układów graficznych.

Wraz z komputerem użytkownik otrzymuje pięć dysków systemowych: Workbench, Extras, Locale, Fonts i Storage. Rozwinięte zostały preferencje systemowe i obsługa urządzeń. Nowe oprogramowanie umożliwia lokalizację systemu (język, waluta, strefa czasowa i inne), ułatwia obsługę drukarek PostScript'o-

wych. a dołączony CrossDOS umożliwia wymianę plików z i BM'em. Pozostałe programy zostały dalece udoskonalone - na przykład przy formatowaniu dysków mamy teraz do dyspozycji opcję Directory Caching, znacznie przyspieszającą dostęp do katalogu.

SZYBKOŚĆ

Dzięki 32 bitowej budowie Amiga 1200 jest komputerem dość szybkim, co znacznie podnosi komfort pracy. Wydajność procesora wynosząca około 3800 Dhrystone/s stawia ją na równi z komputerami opartymi o Intel'a 386SX, a gdy uwzględnimy zyski uzyskiwane dzięki temu, że blitter ma aktualnie znacznie łatwiejszy dostęp do pamięci i porównamy prędkość działania programów na Amidze z ich odpowiednikami pod Windows'ami to okaże się, że A1200 nie ustępuje komputerom typu 386DX!

Aby w pełni wykorzystać moc procesora i jeszcze przyspieszyć działanie, konieczne jest wyposażenie komputera w fast-RAM, a zainstalowanie koprocesora przyspieszy programy korzystające z niego o kolejne kilka - kilkanaście razy. Także wykorzystanie wersji programów dla procesora 68020+ (np. PC-Task) daje około dwukrotny zysk.

Testy przeprowadzone programem AIBB 4.65 na A1200 bez fast-RAM'u wykazują średnie przyspieszenie 2.5 raza w stosunku do A500. Także testy praktyczne potwierdzają tę wartość (PowerPacker, Imploder, Real3D, ADPro, Asm-One) i jest ona praktycznie stała dla różnych programów. A1200 jest tylko 2 razy wolniejsza od A3000 z zegarem 25 MHz, a dzięki nowym chip'om efekt spowalniania programów pracujących w chip-RAM'ie zauważalny jest dopiero przy włączonych ośmiu bitplane'ach w wysokiej rozdzielczości!

Szybszy procesor nie sprawia problemów z kompatybilnością chyba, że korzystamy z bardzo starych i źle napisanych programów. Praktycznie wszystkie programy użytkowe pracują bez problemów, a nawet lepiej. W wypadku gier konieczne jest zwykle wyłączenie pamięci cache i przełączenie chip'ów w tryb ECS - dokonuje się tego w rozbudowanym boot-menu, gdzie można także włączyć NTSC.

SŁOWO NA KONIEC

Możliwości graficzne oferowane przez AA-chipset są już wykorzystywane przez szereg programów, takich jak ADPro, Scala MM200, ProWrite 3.3, czy Brillance (nowy program graficzny firmy Digital Creations). Zapowiadane są nowe wersje Directory Opus'a, Imagine'a, PageStream'a i DeluxePaint'a, które korzystać będą z nowych trybów graficznych i funkcji systemu operacyjnego.

Należy się też spodziewać wkrótce gier używających 256 kolorów i szerokich na 64 piksele sprite'ów. Wyposażenie A1200 w dysk twardy, fast-RAM i koprocesor daje wydajny system, idealny do wszelkich prac graficznych. Amiga 1200 jest dedykowana wszystkim tym, którzy chcą korzystać z 24 bitowej palety kolorów, a nie potrzebują szybkości oferowanej przez A4000, lub jest ona dla nich po prostu za droga.

Konrad Dubiel

JAK ZROBIC WŁASNY TURBOLOADER

czyli programowanie stacji 1541/71 część 1

Każdy chyba wie do czego służy stacja dysków. Nie każdy jednak potrafi w pełni wykorzystać możliwości, jakie projektanci dali użytkownikom stacji Commodore. Prawdopodobnie wynika to z braku fachowej literatury dotyczącej budowy i programowania tych stacji na naszym rynku. Postaram się zapełnić tą lukę tymi artykułami. Stacja dysków zostanie omówiona praktycznie od podstaw, od logicznego podziału dyskietki poczynając, poprzez sterowanie stacji z poziomu języka BASIC, kończąc na zaawansowanym programowaniu w assemblerze. Celem tego cyklu jest takie poprowadzenie Czytelnika (nawet tego, który stacji nigdy nie programował), aby po jego zakończeniu każdy, kto wytrwale starał się zrozumieć podane informacje teoretyczne, oraz wpisywał, uruchamiał i próbował modyfikować podane przykłady był w stanie napisać program przyspieszający ładowanie (ang. turboloader) o żądanych parametrach, czy też inny program rozszerzający DOS stacji. Postaram się w przystępny sposób wyjaśnić większość zagadnień związanych z odczytem, dekodowaniem i przesyłaniem bajtów czytanych z dysku do pamięci operacyjnej komputera. Omówione zostaną praktycznie wszystkie stosowane obecnie metody przyspieszania odczytu, wiele rozwiązań przesyłania programu do stacji, odczytu sektorów, przesuwania głowicy na żądaną ścieżkę, oraz transmisji czytanego programu do komputera. Artykuły te przeznaczone są dla wszystkich, pragnących poszerzyć swe wiadomości w zakresie programowania stacji 1541/71.

Dla pełnego ich zrozumienia jednak, niezbędna jest znajomość zasad programowania w języku maszynowym 6502. Jest to spowodowane tym, że pisanie swoich procedur obsługi stacji jest możliwe tylko w tym języku. Ułatwieniem dla czytelnika będzie posiadanie programu, umożliwiającego monitorowa-

nie pamięci stacji. Programy tego typu zainstalowane są w modułach typu ACTION, czy FINAL 3. Może to też być samodzielny program spełniający to zadanie. Jeszcze dzisiaj postaramy się napisać taki program w BASIC-u i praktycznie powinien on nam wystarczyć. To tyle wstępu. Dziś zajmiemy się podstawami programowania stacji. Omówimy budowę dysku, poznamy także kilka ważniejszych rozkazów dyskowych, i wypróbujemy ich działanie na przykładach. Zaczniemy jednak od budowy dyskietki. Od budowy logicznej oczywiście, czyli przyjrzymy się bliżej temu, czego nie widać.

Zacznijmy od postawienia banalnego pytania. Czym różni się dyskietka od taśmy magnetycznej? Przecież tak na jednym, jak i na drugim można zapisywać dane. Co jednak spowodowało, że tyle ludzi porzuciło niewygodne magnetofony i zaczęło używać dysków? Wadą zapisu taśmowego było to, że po odczytaniu pewnych danych, aby dane te odczytać ponownie, należało taśmę odpowiednio przewinąć, co zwykle zabierało czas. Mniej, lub więcej, ale zawsze!

Wymyślono więc taką taśmę, której przewijać nie trzeba. Zwinięto ją w pętlę, obok niej położono większą pętlę, obok jeszcze większą, a wszystkim tym zaczęto obracać. I tak oto powstała dyskietka. Tak na marginesie, pamięci taśmowe nie wyszły całkiem z użycia. Są one niezastąpione tam, gdzie zachodzi konieczność magazynowania potężnych danych. Tam, gdzie pojemności rzędu gigabajtów są niewystarczające. Dlaczego akurat taśmy? Bo na taśmie zapisać można znacznie więcej niż na dysku. Mowa oczywiście o profesjonalnych urządzeniach do zapisu taśmowego, tak zwanych streamerach.

Dla przykładu porównajmy, ile zmieścić się może na normalnym dysku 5.25", a ile na typowej kasecie fonicznej na przykład C-60. Przyjmijmy, że miarą pojemności będzie powierzchnia nośnika magnetycznego. A więc do dzieła. Najpierw dyskietka. Przyjmimy dużą średnicę pierścienia, na którym dokonuje się zapis: 125mm, a małą: 60mm. Użyteczna powierzchnia jednej strony wyniesie więc:

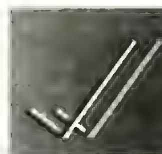
`(PRINT ^*125^2-^*60^2)` 37777 mm^2. Czyli dwie strony, to 75554 mm^2. A teraz taśma. Strona trwa 30 minut, standardowa prędkość przesuwu to około 4.5 cm/s, czyli taśmy w kasecie jest około 81000 mm. Taśma jest 4mm szeroka, a więc mamy 324000 mm^2 powierzchni. Teoretycznie więc na taśmie można 4.5 razy więcej zapisać niż na dysku. A gdyby przyjąć taśmę wielokrotnie szerszą i wielokrotnie dłuższą? Lepiej nie porównywać.

Dysku niestety powększyć raczej nie można. Można co najwyżej zwiększyć upakowanie danych na jednostce powierzchni. ale dotyczy to w takim samym stopniu pamięci taśmowej. No, wystarczy już tego. Przecież mieliśmy zajmować się budową logiczną dysku. Zastanówmy się, jak można coś zapisać na wirującym krążku. Można próbować zapisu spiralnego, takiego, jak na płytach, ale jak zrobić, aby głowica odczytująca nie "zjeżdżała" z zapisanego śladu? Na płycie nie ma tego problemu, igła porusza się w rowku. Ale przecież powierzchnia dysku jest gładka. Istnieją konstrukcje, które umożliwiają śledzenie śladu magnetycznego, ale czy opłaca się je stosować w tego typu stacjach? Czy nie ma innej metody?

Okazuje się, że jest. Jeżeli głowica się nie porusza, a dysk się kręci, to ślad pozostawiony przez głowicę jest okręgiem. Gdyby głowicę przesunąć trochę obok, to otrzymalibyśmy drugi okrąg. Jeszcze dalej trzeci, czwarty itd. Właśnie w ten sposób zapisują wszystkie stacje dysków. Odstęp między tymi okręgami (będziemy je nazywać ścieżkami, ang. track) jest wartością ściśle ustaloną i identyczną we wszystkich stacjach 5.25", które zapisują 40 ścieżek na stronie dysku (np. format IBM 360 kB). Dla ścisłości dodać należy, że stacje 1541/71 pomimo, że potrafią przesunąć głowice na każdą z 40 ścieżek faktycznie zapisują tylko 35 ścieżek na stronie.

Firma Commodore argumentuje to w ten sposób, że wyższe ścieżki są za krótkie, aby wystarczająco pewnie zapisać dane (logiczne: im większy numer ścieżki, tym bliżej środka dysku, a więc i ścieżka krótsza). Należy zauważyć, że przy stałej prędkości wysyłania bitów "przez głowicę", na różnych ścieżkach dane zapisane zostają z różną gęstością (tyle samo danych na różnej długości ścieżkach). Firma Commodore zauważyła, że na niższych ścieżkach zmieścić się może więcej danych i wprowadziła sposób zapisu, który określany jest czasem jako "egzotyczny". Każda strona dysku podzielona została na 4 strefy, z których każda zapisywana jest z inną prędkością.





Wyrównuje to częściowo różnicę gęstości zapisu, a co za tym idzie pozwala na ścieżkach niższych zapisać więcej. Zajmijmy się teraz pojedynczą ścieżką. Przy odczycie stacja musi jakoś rozpoznać, w którym miejscu rozpoczyna się zapis. Podzielono więc ścieżkę na kilka części, tzw. sektorów (ang. sector). Każdy sektor zaś rozpoczyna się od nagłówka (ang. header). Gdy więc stacja potrzebuje odczytać jakieś dane, przesuwa głowicę na odpowiednią ścieżkę, czeka, aż "przyjedzie" nagłówek żądanego sektora i czyta bajty. Sektory w formacie Commodore mają zawsze stałą długość równą 256 bajtów (liczba ta określa przy okazji długość jednej strony pamięci komputera: 256=2^8=$100, gdzie $ oznacza liczbę szesnastkową).



Tak, wszystko to jest sensowne, logiczne, ale... aby odczytać z dysku jakiś program nikt nie będzie się przecież "bawił" w odczytywanie pojedynczych sektorów i składanie tego razem. Od czego w końcu mamy komputer... Okazuje się, że do takiego samego wniosku doszli konstruktorzy w firmie Commodore i zainstalowali w pamięci ROM stacji dysków specjalny program, nazywany Dyskowym Systemem Operacyjnym (w skrócie DOS od Disk Operating System), który wykonuje za nas całą "brudną robotę", dzięki czemu użytkownika wcale nie musi interesować gdzie dokładnie na dysku znajduje się jego program, na ścieżce pierwszej, czy może trzydziestej? Wystarczy, że poda jego nazwę, a stacja sama go znajdzie.

Czy jednak stacja jest na tyle inteligentnym urządzeniem, aby się tego "domyślić"? Oczywiście, że nie. Na dysku zapisywanych jest cały szereg dodatkowych informacji, które służą wyłącznie temu, aby użyt-

kownikowi wygodniej się pracowało. W przypadku stacji Commodore jest to około 3.5% całkowitej pojemności dyskietki. Nie jest to więc zbyt dużo bajtów. Zajmijmy się teraz najpopularniejszą metodą przechowywania danych na dyskach, czyli zbiorami (czasem nazywa się je plikami, ang. file). Jedyną rzeczą, jaką użytkownik musi pamiętać to nazwa zbioru. Gdy poślemy ją do stacji, której polecimy odczytać zbiór, urządzenie to najpierw przegląda spis wszystkich programów zapisanych na tym dysku, nazywany dalej katalogiem dysku (ang. directory). W przypadku nie znalezienia wpisu o dokładnie takiej samej nazwie jak podana, stacja odpowiada, że na tym dysku nie ma takiego zbioru (ang. file not found).

Gdy jednak poszukiwania zakończą się sukcesem, stacja ma "do wglądu" wszystkie dane o tym zbiorze, np. jaki jest długi, jakiego jest typu, czy jest zabezpieczony przed zapisem, gdzie się zaczyna i wiele innych informacji. Przy odczycie szczególnie ważna jest informacja, gdzie na dysku znajduje się początek zbioru. Jest ona zapisana w postaci numeru ścieżki i numeru sektora. Zapis taki jednoznacznie określa w jakim sektorze znajdują się pierwsze bajty zbioru. Nic nie stoi więc na przeszkodzie, aby sektor ten odczytać. I co dalej? Gdzie znajduje się dalszy ciąg zbioru? Firma Commodore rozwiązała ten problem w ten sposób, że wprowadziła tak zwane linki.

Polega to na tym, że w każdym sektorze zapisanych jest tylko 254 zamiast 256 bajtów danych, a w dwóch pierwszych bajtach każdego sektora zapisany jest numer ścieżki i sektora (tzw. link), gdzie znajduje się dalszy ciąg aktualnie czytanego zbioru. Jeżeli dojdziemy tak do sektora ostatniego, to nie jest już potrzebny wskaźnik następnego bloku danych (określenie "blok" stosuje się zamiennie z określeniem "sektor"). Dwa pierwsze bajty zawierają więc informację, że jest to ostatni sektor zbioru, oraz ile bajtów tego sektora jeszcze do zbioru należy (długości zbiorów nie muszą być przecież wielokrotnością liczby 254).

Tak właśnie stacje Commodore zapisują wszystkie zbiory sekwencyjne, czyli zbiory z rozszerzeniem SEQ, PRG i USR. Z punku widzenia stacji zbiory te niczym nie różnią się między sobą. Przeważnie przyjmuje się, że zbiory typu PRG w dwóch pierwszych bajtach (w dwóch pierwszych bajtach zbioru oczywiście, czyli w trzecim i czwartym bajcie pierwszego sektora) zawierać powinny tzw. load-address, czyli adres w pamięci komputera, gdzie zbiór taki powinien być załadowany (na przykład zbiór z programem w języku assemblera, który może prawidłowo działać tylko w tym miejscu, gdzie został napisany).

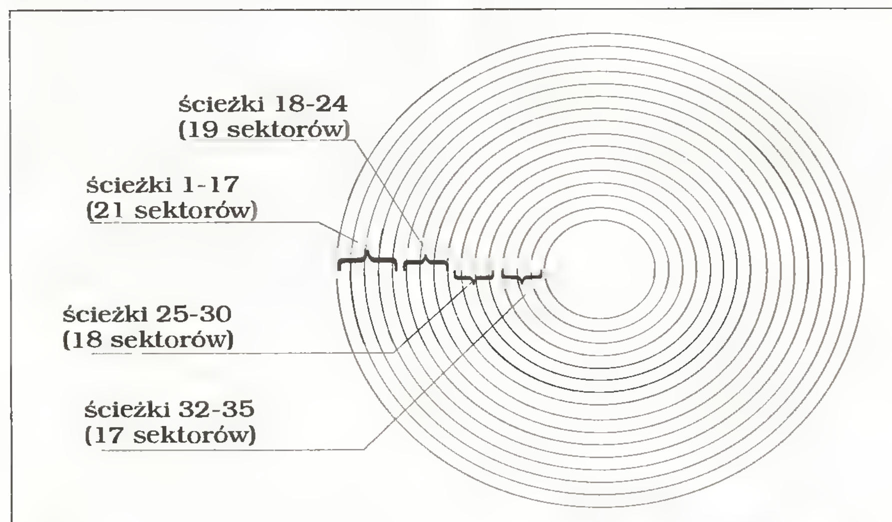
Nie jest to jednak regułą. Czasem stosuje się zbiory PRG, które są w istocie normalnymi zbiorami typu SEQ (różnią się tylko nazwą typu). Dla stacji nie jest jednak istotne, czy zbiór PRG ma load-address, czy też nie. To wyłącznie sprawa komputera, jak potraktuje dwa pierwsze bajty zbioru. Stacje Commodore umożliwiają także zapis zbioru w zupełnie inny sposób (zbiory typu REL). Jest on dużo bardziej skomplikowany od zapisu sekwencyjnego i nie będziemy się nim na razie zajmować. Zajmiemy się za to czymś konkretniejszym, a mianowicie rozkazami bezpośredniego dostępu (ang. direct access commands). Cóż to takiego?

Stacje 1541/71 posiadają zbiór rozkazów pozwalających na dostęp do danych w pojedynczych sektorach dysku. Używając tych rozkazów można stworzyć własne systemy przetwarzania danych, rezygnując z "usług" DOS-u. Oczywiście, wymaga to włożenia znacznie więcej pracy, niż przy korzystaniu z plików sekwencyjnych, czy relatywnych. Proponuję używać czystej dyskietki w czasie eksperymentów, aby uniknąć przypadkowego utracenia jakichś ważnych danych. Wiemy już, że dyskietka jest zorganizowana w ścieżki i sektory, oraz, że ścieżki zewnętrzne są dłuższe od wewnętrznych, przez co może się na nich zmieścić więcej sektorów.

Scieżki są ponumerowane od ścieżek zewnętrznych poczynając, na wewnętrznych kończąc. Ścieżka zewnętrzna ma więc numer 1 i zawiera najwięcej sektorów. Ścieżka wewnętrzna, to ścieżka 35. Wiemy też, czemu nie są używane ścieżki powyżej 35. I w tym miejscu pojawia się pierwsza różnica pomiędzy stacją 1541, a 1571. Jak się później okaże, różnic tych będzie znacznie więcej. Stacja 1541 ma jedną głowicę odczytująco - zapisującą. Wynika z tego, że stacja może zapisywać tylko jedną stronę dysku. Ale przecież wszystkie dyskietki mają nośnik magnetyczny po obu stronach. Czy można więc pozwolić, aby "marnowała się" połowa dyskietki?

Użytkownicy 1541 radzą sobie z tym problemem w ten sposób, że wycinają z drugiej strony dysku dodatkowy otwór, informujący, że na dyskietkę można zapisywać i używają dyskietki ponownie, wkładając ją do stacji "do góry nogami".

Na 1571 jest z tym trochę problemów. Stacja ta ma dwie głowice, jedną z góry, jedną z dołu. Aby, dla wyspecyfikowania konkretnego sektora nie trzeba było podawać trzech informacji (strona, ścieżka, sektor) przyjęto, że ścieżka na drugiej stronie odpowiadająca ustawieniu głowicy do odczytu ścieżki pierwszej ma numer 36. Jak łatwo się domyślić, podając numery ścieżek



w zakresie 1-70 mamy dostęp do danych zapisanych na obydwóch stronach dysku. Czyżby więc na 1571 można było czytać dyski zapisane z dwóch stron na 1541 bez obracania strony? Ależ nie!

Przecież jeżeli włożymy do stacji dysk "do góry nogami" to kręci on się w przeciwną stronę, niż włożony normalnie! W jaki więc sposób stacja miałaby odczytać sektory, jeżeli najpierw "przyjechałby" koniec, a później początek, a w dodatku wszystkie bajty byłyby odwrócone? Poza tym dwie strony na 1541 są traktowane tak, jak dwa oddzielne dyski, na 1571 natomiast, jak jeden dysk, ale o dwa razy większej pojemności (możemy np. zapisać jeden zbiór o długości 1328 bloków, czego nie uda nam się zrobić na 1541). Chciałbym w tym miejscu zwrócić uwagę na używane określenia. Niektórzy, jako dysk dwustronny uważają dysk 1541, a niektórzy 1571. Poprawniejsze jest określanie mianem dysku dwustronnego dysku 1571, ale nie wszyscy się do tego stosują i prowadzi to do nieporozumień.

Niektórzy mówiąc "dysk dwustronny" mają na myśli dwa oddzielne dyski logiczne 1541 zapisane po obydwóch stronach jednego dysku fizycznego. Aby takich nieporozumień uniknąć postaram się stosować określenia dysku 1571 i dysku 1541. Jak dokładnie wygląda rozmieszczenie sektorów na ścieżkach najlepiej widać w poniższej tabelce:

1-17 / 36-52	 0-20
18-24 / 53-59	 0-18
25-30 / 60-65	 0-17
31-35 / 66-70	 0-16

Jeżeli możemy skłonić stację dysków do odczytania sektora z dyskietki, to wyrasta kolejne pytanie, co z nim dalej zrobić. Ponieważ pojedyncze znaki odczytywane są z prędkością ponad 38000 bajtów na sekundę nie ma szans na przesłanie ich w czasie rzeczywistym do komputera. Sektor musi być czasowo przechowany tak, aby można go było przetwarzać nawet używając BASIC-a. Identycznie jest przy zapisie sektora na dyskietkę. Stacja 1541/71 ma pięć buforów, każdy o długości 256 bajtów. Pamięć ta używana jest przy ładowaniu programów, pracy z plikami itp. Powróćmy jednak do rozkazów bezpośredniego dostępu. Przed dostaniem się do dowolnego sektora należy zarezerwować dla siebie bufor poleceniem:

```
OPEN FILE,DEVICE,CHANNEL,"#BUFFER"
```

gdzie "file" jest numerem zbioru, "device" numerem urządzenia (przeważnie mamy do czynienia z urządzeniem numer 8), "channel" to numer kanału dyskowego, a "buffer" to numer bufora stacji dysków (z przedziału 0-3). Znak "#" podany jako nazwa pliku mówi stacji, że ma być otworzony kanał bezpośredniego dostępu, do którego stacja dołącza następnie bufor o podanym numerze. Numer ten zwykle można pominąć. Stacja w takiej sytuacji automatycznie wybierze któryś z wolnych buforów. Kiedy samemu wybiera się bufor trzeba mieć pewność, że nie jest on wykorzystywany w innych celach.

Jeżeli wszystkie bufory, lub ten wymagany są już przypisane do innych zadań stacja zakomunikuje: 70 NO CHANNEL (brak wolnego kanału). Spróbujmy teraz odczytać jakiś sektor. Rozkaz BLOCK-READ ma następującą składnię:

```
"U1:CHANNEL,DRIVE,TRACK,SECTOR"
```

gdzie "channel" to ten sam numer, jaki podaliśmy przy otwieraniu zbioru, "drive" to numer napędu (stacje 1541 i 1571 są jednonapędowe, a więc dla stacji tych numer ten może mieć tylko wartość 0), a "track" i "sector" to numery ścieżki i sektora, które nie wymagają chyba komentarza. Wszystkie parametry mogą być oddzielone przecinkami lub znakami spacji. Słowo rozkazowe (np. U1) może być oddzielone od parametrów dwukropkiem, lub znakiem spacji.

A teraz przykład:

```
10OPEN 1,8,15
20OPEN 2,8,5,"#"
30PRINT#1,"U1:5,0,18,0"
40FOR A=0 TO 255
50GET#2,a$
60POKE 1024+A,ASC(A$+CHR$(0))
70NEXT 80 CLOSE 2:CLOSE 1
```

Programik ten odczytuje podany sektor z dysku. Kolejne bajty sektora pojawiają się na górze ekranu. W ten sposób można najszybciej i najprościej obejrzeć zawartość sektora. Można oczywiście gromadzić bajty w jakiejś tablicy, czy magazynować gdzieś w pamięci RAM, ale nam w zupełności wystarczy "wyrzucenie" bajtów na ekran (pamięć ekranu rozpoczyna się od adresu 1024). Program ten nie testuje żadnych błędów. Nie będzie on działał prawidłowo, gdy np. nie będzie wolnych buforów, nie będzie włożonego dysku, stacja będzie wyłączona, itp. Sprawą błędów zajmiemy się kiedy indziej. Poznajmy teraz dwa kolejne polecenia. Pierwsze, to zapis sektora:

```
"U2:CHANNEL,DRIVE,TRACK,SECTOR"
```

Czy ta składnia czegoś nam nie przypomina? Jest identyczna, jak rozkazu czytającego sektor. Nie ma potrzeby więc jej opisywać. Poznajmy drugi rozkaz (BUFFER-POINTER):

```
"B-P:CHANNEL,BYTE"
```

Rozkaz ten ustawia wskaźnik bufora przyłączonego do podanego kanału na bajt określony parametrem "byte", który określa pozycję w buforze, od której nastąpi odczyt, lub zapis. A teraz pora na jakiś użyteczny program. Używając poznane rozkazy zmienimy nazwę naszego dysku (pomimo, że DOS nie zawiera polecenia służącego do tego celu):

```
10 OPEN 1,8,15 15 OPEN 2,8,5,"#"
20 PRINT#1,"U1:5,0,18,0"
25 PRINT#1,"B-P:5,144"
30 N$=""
35 FOR I=1 TO 16
40 GET#2,A$:n$=N$+A$
45 NEXT
50 PRINT"aktualna nazwa dysku:"N$
55 INPUT"nowa nazwa dysku";N$
60 N$=N$+"                ":REM 16 spacji
65 PRINT#1,"B-P:5,144"
70 PRINT#2,LEFT$(N$,16);
75 PRINT#1,"U2:5,0,18,0"
80 PRINT#1,"I"
85 CLOSE 2:CLOSE 1
```

Wykonajmy powyższy program, po czym wpiszmy LOAD"$",8 a następnie LIST. I co, zmieniła się nazwa? Jeżeli nie popełniliśmy żadnego błędu, ani nie wystąpił żaden błąd dyskowy, to powinna. Kazaliśmy przecież stacji odczytać zawartość sektora 18,0 (czytaj: sektora 0 na ścieżce 18), wpisać nową nazwę od 144 bajtu bufora poczynając (tam właśnie znajduje się nazwa dysku), oraz zapisać z powrotem ten sektor na dysk. Inicjowanie dysku (linia 80) było konieczne, aby stacja zauważyła zmianę nazwy dysku.

W przeciwnym wypadku w katalogu pojawiłaby się stara nazwa, pomimo, że dysk dawno już tak się nie nazywa. Jest to spowodowane tym, że stacja w jednym z buforów przechowywuje kopię sektora 18,0 którą uaktualnia po każdorazowym inicjowaniu dysku. Zamiast inicjować dysk można go było wyjąć i ponownie włożyć do stacji. Inicjowanie odbyłoby się wtedy automatycznie przed odczytem katalogu. Napiszemy teraz jeszcze jeden programik, który może kiedyś nam się przydać. Najpierw jednak poznamy jeszcze jedno polecenie, służące do pobierania bajtów z dowolnego miejsca w pamięci stacji (MEMORY-READ):

```
"M-R"+CHR$(lo)+CHR$(hi)+CHR$(number)
```

Jak widać, przed rozkazem tym nie trzeba otwierać żadnego kanału, jest więc on łatwiejszy w użyciu. "lo" i "hi" to młodszy i starszy bajt adresu w stacji, skąd chcemy pobierać bajty. "number" to ilość bajtów jaką chcemy pobrać. Tak więc: 1 oznacza 1 bajt, 255 - 255 bajtów, a 0 (uwaga!) 256 bajtów, dzięki czemu można na raz pobrać cały blok. A oto program, który przeniesie całą pamięć RAM stacji (2 kB) do pamięci operacyjnej komputera od adresu $2000 do $2800:

```
10 OPEN 1,8,15
20 FOR B=0 TO 7:PRINT B
30 PRINT#1,"M-R"CHR$(0);
CHR$(B)CHR$(0)
40 FOR A=0 TO 255
50 GET#1,A$
60 POKE 8192+256*B+A,
ASC(A$+CHR$(0))
70 NEXT A,B
80 CLOSE 1
```

Program ten dość długo się wykonuje, ale spełnia swoje zadanie. Zachęcam do eksperymentowania w tej dziedzinie, do wprowadzania zmian w programach i obserwowania skutków. Następnym razem zajmiemy się już sprawami poważniejszymi (chociaż tak samo łatwymi), czyli przejdziemy do programowania w assemblerze.

CO TO JEST "KEBAB"?

Czy wiecie co to jest Kebab? To takie coś w zestawie obok hamburgerów itp. no nie? NIE! Dzięki spostrzegawczości Pana Andrzeja Graczyńskiego (pozdrowienia od Redakcji!) wiemy już, że Kebab to stara potrawa arabska.

Przyrządzenie Kebaba polegało na upieczeniu nad ogniem surowych, nie przyprawionych kawałków mięsa, ponadziewanych osobno na oddzielnych gałązkach większej, umieszczonej nad ogniem gałęzi. Opis znaleźć można w książce Karola Maya "Karawana Niewolników".

Dziękujemy!

Szanowna Redakcjo KEBAB'a

Piszę ten list jako odpowiedź na zarzuconą mi winę. Na łamach KEBAB'a 7-8'92 przeczytałem artykuł w którym życzy mi się robaków w komputerze i przerw w zasilaniu. Pragnę usprawiedliwić moje postępowanie przy tworzeniu wirusa NEMESIS. Nie jest on w swoim założeniu wirusem złośliwym, jego celem jest zasygnalizowanie problemu zbyt dużego zaufania dla prostych bootblock'ów antywirusowych typu Mutand LED itp. Jedyna złośliwość mojego wirusa polega na wieszaniu systemu po około 1 godzinie pracy, dodatkowo 1 minutę przed powieszeniem wirus sygnalizuje odliczanie czasu wstecz poprzez coraz szybsze miganie diody POWER. Myślę, że jest to dobry kompromis pomiędzy złym wirusem, a zaznaczeniem w/w problemu. Pragnę przedłożyć jedno sprostowanie do mojego wirusa. W artykule napisano, że aby wykryć i usunąć NEMESIS wirusa potrzebny jest jakikolwiek bootblock nie odwołujący się do Display Alertu. Nie jest to prawda, gdyż wirus ten ominie większą część takich antywirusów (proszę to dokładniej sprawdzić). Istota omijania BB nie opiera się tylko na przechwyceniu wektora adresu Display Alertu. Dlatego osobiście polecam NEMESIS protektora, który został stworzony w celu wykrywania tego wirusa. Na koniec pragnę przeprosić wszystkich, którym mój wirus wyrządził "szkody moralne, jak i materialne".

Z poważaniem

BlackBird aktualnie z INF Szczecin

Errata do numerów 10' 92 oraz 11-12' 92

Z przykrością stwierdzamy, że "Errata do poprzedniego numeru" stała się naszą comiesięczną rubryką, zatem i ostatni numer nie jest tu wyjątkiem. Tym razem poprawki dotyczą listingu ze strony 35. Linijkę skomentowaną "ustawiamy kolory tła i tekstu" której treść brzmi:

```
dc.l   $01800000,$01800fff
```

należy zmienić na:

```
dc.l   $01800000,$01820fff
```

Również w numerze 10-tym, na stronie szóstej mamy dwie linijki listingu:

```
A5026 LDA $D012
A5029 BNE $5025
```

Otóż druga z nich powinna oczywiście wyglądać następująco:

```
A5029 BNE $5026
```

ADPro dla "ograniczonych".

Praktycznie każdy szanujący się program do ray-tracingu posiada opcję generacji grafiki 24-bitowej, gdzie każdy piksel znajdujący się na ekranie może przybrać dowolny kolor z palety 16 777 216 barw. Jakość takiego obrazu jest naprawdę doskonała, a ludzkie oko, ze swoją zdolnością do odróżniania zaledwie 400000 ocieni nie jest w stanie doszukać się żadnych nieciągłości w przejściach barwnych.

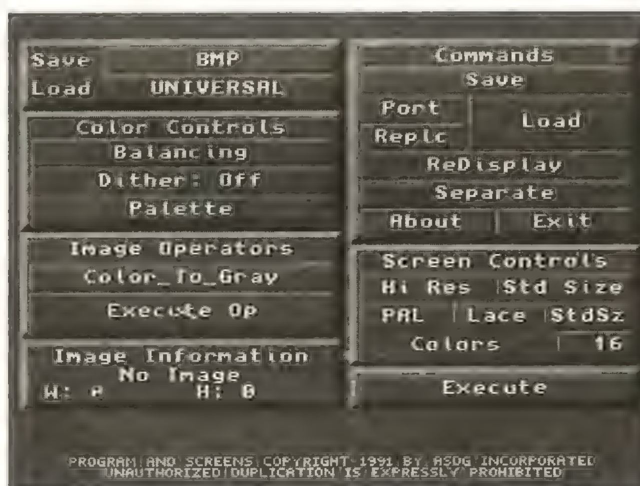
Paletą taką dysponują nowe Amigi wyposażone w AA chipset, w Polsce obecne w niewielkiej niestety liczbie. Co więc począć mają właściciele tych "normalnych" Amig? Programy do ray-tracingu owszem, potrafią tworzyć również grafikę wykorzystującą HAM (który tak na marginesie tylko przypadkiem uchował się w Amidze, jako że jeszcze na krótko przed premierą nowego komputera wydawał się konstruktorom zbyt zawiły, aby wzbudzić zainteresowanie producentów oprogramowania i sprzętu), lecz jest ona na ogół wysoce niedoskonała. Przyczyny?

Ze specyfiki trybu HAM wynika natychmiastowy dostęp do jedynie szesnastu kolorów spośród amigowych 4096, pozostałe barwy trzeba tworzyć przez mozolne operowanie poszczególnymi składowymi koloru (czerwoną, zieloną i niebieską). Kluczowym zagadnieniem jest zatem właściwy dobór owej podstawowej, szesnastokolorowej palety. Ray-tracery (zresztą programy do malowania również) mają raczej mgliste pojęcie o barwach jakie pojawią się na docelowym obrazie - stąd niskiej jakości HAM'owe obrazki z programu Real 3D, który przecież w przypadku używania karty 24-bitowej daje efekty niemalże fotograficzne.

Dobrym sposobem poradzenia sobie z problemem jest wygenerowanie grafiki najpierw w pełnym spektrum (nie, nie ZX-48K!), a dopiero potem przeliczenie jej na rozdzielczość i paletę docelową. Ma to jeszcze tę zaletę, że konwerter dokonujący dzieła będzie mógł używać techniki roztrząsania (ang. dithering - sztuczne tworzenie nowych barw poprzez ustawianie obok siebie pikseli różnych kolorów).

No dobrze, ale jaki konwerter? Art Department Pro naturalnie! Mimo wielokrotnie pojawiających się w różnych magazynach sugestii, że bez wyposażenia naszej Amigi w trzy hektary RAMu, nie mamy nawet co patrzeć na ten program, ja osobiście używam go z dobrym skutkiem na Amidze z 1M RAM i jedną stacją dysków i choć muszę przyznać, że mogłoby być lepiej z wielkością bufora oferowanego mi przez program, to jednak daleki jestem od załamania.

Jak zatem przygotować ADPro do pracy? Przede wszystkim należy zdearchiwizować zawartość zbiorów *Operators.lzh, Savers.lzh i Loaders.lzh* znajdujących się na dyskietkach dystrybucyjnych. W wyniku otrzymamy kilkadziesiąt plików, które należy teraz powtórnie skompresować (w przeciwnym wypadku nie zmieściłyby się na jednej dyskietce) z tym, że z użyciem Imploder'a (PowerPacker odpada - a dlaczego, przeczytasz w poprzednim Kebabie), przy czym sugerowałbym uruchomienie go z opcją SHORTROOT. Sam ADPro również kompresujemy Imploderem i całość zgrywamy na jeden dysk, umieszczając operator'y, loader'y i saver'y odpowiednio w katalogach: **Operators2, Loaders2 i Savers2.** Wyjątkiem jest tu zbiór *.Edit_Palette*, który musi znajdować się w tym samym katalogu co główny plik ADPro.



Muszę od razu powiedzieć, że nie wszystko się Wam zmieści. Z drugiej jednak strony nie widzę sensu w upychaniu na przykład zbioru *Text_visual*, który i tak na jednym megabajcie zadziałać nie zechce.

Gdy już zdecydujemy się, jakie funkcje są nam niezbędne, a jakie - nie, przystępujemy do ostatniej fazy - uruchomienia programu. Nie, nie wystarczy napisać nazwy programu. Firma ASDG bardzo wcześnie miała informację o nowym (AA) chipsecie, w związku z czym zainstalowała w swoim produkcie funkcje do obsługi nowych trybów graficznych i rozszerzonej palety. Funkcje te ładują się standardowo na każdej Amidze, pożerając niepotrzebnie spory obszar pamięci RAM. Jeśli więc ich nie potrzebujecie, w *startup-sequence* napiszcie:

```
ADPro NE
```

...i macie do dyspozycji bufor o rozmiarze ponad 347KB, co jest wystarczające do załadowania, obróbki i wyświetlenia kolorowego (a więc także 24-bitowego) obrazka w rozdzielczości 320x256 punktów.

Do następnego programu zatem,

Mirosław "Thorgal" Smyk



TURBO ASSEMBLER V5.1

Jednym z popularniejszych assemblerów na Commodore 64 jest bez wątpienia *TURBO ASSEMBLER V5.1* firmy *OMICRON SOFTWARE*. Zarówno pod względem komfortu edycji jak i szybkosci asemblacji jedynie *MacroAssembler* firmy *Heinz Heisse Verlag* może mu dorównać. Mimo, że Turbo Assembler jest dość powszechne używany,to jednak tylko niewiele osób zna cały repertuar możliwości jakimi dysponuje.

Zacznijmy może od komend związanych z edycją. Wszystkie wywołujemy w następujacy sposób: wciskamy klawisz oznaczony "<-" (strzałka w lewo) a następnie klawisz z literą odpowiadającą danej komendzie.

Oto ich kompletna lista:

1 - Powrót do BASIC'a lub monitora (w zależności od sposobu uruchomienia assembler'a)

2 - Wstawienie do tekstu programu linii komentarza składającej się z poziomych kresek. Może ona np.oddzielać poszczególne procedury itp.

3 - Asemblacja kodu do pamięci.

4 - Wydrukowanie listingu programu. Możemy podać następujące parametry:

? - drukarka, **'** - ekran lub nazwę pliku, co spowoduje utworzenie pliku tekstowego na dysku.

5 - Asemblacja na dysk (nie zmienia zawartości pamięci).

6 - Transfer danych z pamięci do kodu źródłowego (tworzy linię z komendą .BYTE!).

7 - Ustawia tabulację na kolumnę w której znajduje się kursor (jest to kolumna na której kończą się etykiety a zaczynają się rozkazy).

Q-Przesuwa kursor do lewej krawędzi ekranu.

W - Zapisuje kod źródłowy jako plik tekstowy (typu SEQ) możliwy do odczytania przez inne assemblery lub edytor tekstu.

E - Wczytuje kod źródłowy zapisany przez komendę W.

R - Zastępuje podany ciąg znaków drugim ciągiem, który również musimy podać. Znajduje tylko pierwsze wystąpienie danego ciągu. Po czym musimy użyć jednej z dwóch komend:

T - Zamienia pierwszy ciąg

Y - Zamienia wszystkie takie ciągi występujące w tekście.

U - Drukuje wszystkie etykiety. Parametry jak przy komendzie 4.

I - Wypełnia zerami`pamięć od adresu $0800 do początku kodu źródłowego.

O - Zmiana kolorów tła,ramki i znaków.

P - Po wybraniu tej komendy program prosi nas o podanie numeru z zakresu $00-$ff przez tą wartość będzie EOR'owany każdy bajt kodu źródłowego w czasie zapisu i odczytu. Pozwala to na zabezpieczenie naszego programu przed podglądaniem przez ciekawskich.

@- Podaje aktualny status operacji dyskowych (raporty o błędach itp.)

***** - Drukuje katalog dysku.

+ - Pobiera dwie liczby heksadecymalne i podaje ich sumę.

- - Jak wyżej ale podaje różnicę.

^ - Wstawia linię, w której znajduje się kursor do bufora.

**** - Wstawia zawartość bufora do linii, w której znajduje się kursor.

A - Pozwala na wpisywanie kodów kontrolnych. Jest to odpowiednik trybu cudzysłowia w BASIC'u. Przerywamy za pomocą "<-"!

L - Wczytuje Kod źródłowy zapisany jako plik skompresowany (specyficzny format zapisu tego assemblera).

S - Zapisuje kod źródłowy jako plik skompresowany.

D - Umożliwia przesyłanie komend do stacji dysków.

F - Szuka w tekście programu podanej sekwencji. Zatrzymuje się po znalezieniu pierwszego jej wystąpienia.

H - Wyszukuje następne wystąpienie sekwencji (Patrz "F").

G - Przesuwa kursor do linii oznaczonej podanym znacznikiem (1-9 oraz S i E. Znaczniki S i E oznaczają początek i koniec bloku na którym można dokonywać różnych operacji. Patrz"B").

J - Wyświetla na ekranie zawartość pamięci miedzy podanymi adresami (Jak opcja M w monitorze kodu maszynowego).

K - Po naciśnięciu tego klawisza musimy wcisnąć jeden z klawiszy funkcyjnych (F3-F6) i możemy przypisać mu dowolną sekwencję znaków.

: - Wyświetla aktualne ustawienie wszystkich znaczników.

C - Zimny start programu.

V - Orientacyjna mapa pamięci ukazująca wolne i zajęte bloki pamięci.

B - Operacje na blokach (Blok musi być wcześniej zdefiniowany za pomocą komendy M!). Po naciśnięciu tego klawisza musimy nacisnąć jeszcze jeden, który odpowiada wybranej przez nas operacji. Są to:

c - kopiowanie bloku do pozycji kursora

k - skasowanie bloku

w - zapisanie bloku na dysk w postaci pliku tekstowego.

N - Skok do wybranego wiersza w kodzie źródłowym.

M - Ustawienie znacznika w tekście programu. Po wybraniu tej komendy musimy nacisnąć klawisz odpowiadający danemu znacznikowi (Patrz "G").

= - Łączenie linii (tej w której znajduje się kursor i poprzedniej).

RETURN - Rozdzielanie linii (z tekstu znajdującego się po lewej stronie kursora tworzy nową linię).

; - Usuwa podany znacznik.

SHIFT + L - Wczytuje dane z dysku do pamięci.

SHIFT + S - Zapisuje dane z pamięci na dysk.

SHIFT + F - Wypełnianie pamięci z podanego zakresu dowolną wartością.

SHIFT + K - Włączenie i Wyłączenie dzwięku towarzyszącego wciśnięciu klawisza.

F1 - Włączenie i wyłączenie klawiszy funkcyjnych.

DEL - Kasuje linię w której znajduje się kursor.

/ - Kasuje tekst znajdujący się za kursorem.

INST - Włączenie i wyłączenie automatycznego rozsuwania linii po RETURN.

By wstawić do tekstu programu znak "<-" wystarczy wcisnąć ten klawisz dwa razy.

Kolej teraz na komendy wywoływane bez uprzedniego wciśnięcia "<-".

F1 - Przesunięcie kursora o 24 wiersze do góry.

F2 - Przesunięcie kursora na początek kodu źródłowego.

F7 - Przesunięcie kursora o 24 wiersze do dołu.

F8 - Przesunięcie kursora na koniec kodu źródłowego.

Klawisze **F3 - F6** są definiowalne przez użytkownika. Normalnie mają one przypisane następujące funkcje:

F3 - drukuje ".TEXT"

F4 - Asemblacja i automatyczne uruchomienie programu.

F5 - drukuje ".BYTE"

SHIFT + INST - Przełącza tryby wstawiania i kasowania.

I to już wszystkie komendy TURBO ASSEMBLER'a. Myślę, że teraz już każdy będzie mógł wykorzystać wszystkie możliwości tego doskonałego narzędzia.

Krzysztof "BRUSH" Dąbrowski

Mapa pamięci Amigi

- ciąg dalszy

08E DIWSTRT Z A

090 DIWSTOP Z A

Rejestry te ustalają odpowiednio lewy, górny (DIWSTRT) oraz prawy, dolny (DIWSTOP) róg wyświetlanego obrazu. Młodsze połówki tych rejestrów (bity 0-7) odpowiadają za pozycję poziomą, natomiast starsze pozycję pionową. Wszelkie obiekty mające być wyświetlone (zarówno sprite'y jak i Bit-Plane'y) muszą mieścić się w zakresie tego okna; poza nim nie można ich wyświetlić. Odpowiednio dobierając wartości tych rejestrów możemy zdefiniować standardowy tryb wyświetlania (320 punktów, mierzonych w trybie Lo-Res, szerokości), tzw. tryb Overscan likwidujący "bordery", jak i każdą inną wielkość. Wydawać by się mogło, że lewy górny róg (a więc rejestr DIWSTRT) powinien zawierać wartość $0000, czyli pozycję (0,0). Niestety z uwagi na specyfikę układów graficznych, okazało by się, że przy tak zdefiniowanym położeniu obraz "uciekłby" nam z ekranu. Dlatego zazwyczaj za współrzędną pionową przyjmuje się wartość $2C, natomiast za poziomą wartość $81. Tak ustalone współrzędne można ustwiać następującym rozkazem procesora:

```
move.w #$2C81,$DFF08E
```

lub rozkazem Coppera:

```
dc.l $008E2C81
```

W indywidualnych przypadkach wartości te (w szczególności współrzędna pozioma) mogą wymagać zweryfikowania, gdyż ich dopasowanie w dużej mierze zależy od ustawienia posiadanego monitora. Gdy rejestr definiujący początek wyświetlanego okna mamy przygotowany, należy ustawić rejestr DIWSTOP, określający koniec wspomnianego okna. W tym celu jeżeli zdecydowaliśmy się na pracę w oknie o szerokości 320 punktów trybu LoRes i wysokości 256 punktów to wspomniane wymiary należy dodać do ustalonych współrzędnych początku.

$81 + 320dec = $1C1

$2C + 256dec = $12C

Obcinamy najstarszy bit i wpisujemy do DIWSTOP wartość $2CC1:

```
move.w #$2CC1,$DFF090
```

lub Copperem:

```
dc.l $00902CC1
```

Proszę zwrócić uwagę, że rejestry te **NIE** definiują nam rozdzielczości, a jedynie położenie na monitorze (telwizorze) i wielkość wyświetlanego obszaru roboczego.

092 DDFSTRT Z A

094 DDFSTOP Z A

Rejestry te określają miejsce na ekranie od którego DMA będzie pobierało dane z pamięci i powodowałó wyświetlanie. Rejestry te definiują jaką wielkość wyrażoną w słowach będzie miała każda linia (Standardowo dla trybu LoRes 20, HiRes 40). Jak oblicza się wartości dla tych rejestrów?

LORES:

```
DDFSTRT=(DIWX/2)-8.5

DDFSTOP=DDFSTRT+8*(IloscSlow-1)
```

HIRES:

```
DDFSTRT=(DIWX/2)-4.5

DDFSTOP=DDFSTRT+4*(IloscSlow-2)
```

gdzie: DIWX - pozioma pozycja rejestru DIWSTRT.

Przykład: Zakładając ustawienie DIWSTRT jak opisano wyżej ($81) obliczyć DDFSTRT i DDFSTOP dla niskiej rozdzielczości.

```
DIWX = $81

DDFSTRT = ($81/2) - 8.5 = $38

DDFSTOP = $38 + 8 * (20 - 1) =
$38 + 8 * 19 = $D0
```

czyli:

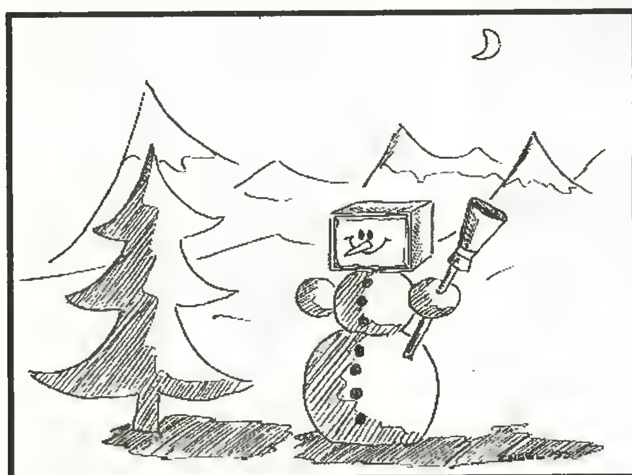
```
move.w #$0038,$DFF092

move.w #$00D0,$DFF094
```

lub

```
dc.l $00920038

dc.l $009400D0
```



096 DMACON Z ADP

Rejestr kontroli DMA. Poszczególne bity posiadają identyczne znaczenie jak w rejestrze DMACONR. Rejestr ten służy do zapisu, czyli włączania/wyłączania poszczególnych kanałów DMA. Znaczenie bitu 15 patrz rejestr INTENA.

09A INTENA Z P

09C INTREQ Z P

Rejestry te odpowiedzialne są za obsługę przerwań. Ich zadanie, oraz znaczenie poszczególnych bitów zostało omówione w artykule "O przerwaniach" w Kebabie 7-8/92. Tutaj chciałbym jedynie szerzej skomentować znaczenie i sposób interpretacji tajemniczego bitu numer 15 (SET/CLR). Jak sama nazwa wskazuje służy on do ustawiania lub kasowania innych bitów tego rejestru. I tak jeżeli ustawimy bit 15, oraz bity np. 3, 5, 12, i tak przygotowaną wartość prześlemy do rejestru INTENA rozkazem:

```
move.w #%1001000000101000,$dff09A
```

to w rezultacie uzyskamy ustawienie właśnie bitów 3, 5, 12. Pozostałe bity nie zostaną zmienione!!! - więc jeśli bit na przykład 2 był ustawiony przed wykonaniem powyższej instrukcji, to dalej zachowa swoją wartość. Jak zatem skasować bity 3, 5 i 12? Proszę bardzo:

```
move.w #%0001000000101000,$dff09A
```

Jak zatem widać wszystkie ustawione bity (z wyjątkiem 15) przyjmują wartość bitu 15, a pozostałe zostają niezmienione. Analogiczny mechanizm ustawiania/kasowania bitów dotyczy wszystkich rejestrów, w których bit o numerze 15 nazywa się SET/CLR.

09E ADKCON Z P

KONTROLER pracy Audio i dysku. Poszczególne bity mają takie same znaczenie jak przy ADKCONR, z tą różnicą, że są przeznaczone dla zapisu. Znaczenie bitu 15 patrz INTENA.

Rejestry 0A0 - 0AA, 0B0 - 0BA, 0C0 - 0CA, 0D0 - 0DA dotyczą czterech kanałów Audio i ich znaczenie i praktyczne wykorzystanie zostało omówione w Kebabie 10/92 w artykule "Świat dźwięku".

Krzysztof Kobus

REALna animacja...

Programów do tworzenia fotorealistycznych obrazów metodą śledzenia promieni (ang. ray-tracing) powstało już dla Amigi tak wiele, że wybór najlepszego spośród nich jest bardzo trudny. Największe możliwości, szczególnie animacyjne, posiada Imagine, lecz jest on dość trudny w obsłudze i powolny. Jego największym konkurentem jest Real3D Professional rozprowadzany przez firmę Activa International. Do niedawna sprzedawana była wersja 1.4 tego programu, lecz istnieje już nowa - 2.0, która jest wyposażona w wiele nowych funkcji, między innymi ułatwiających tworzenie animacji.

Real wyróżnia się spośród pozostałych programów tego typu przede wszystkim budową obiektów. Nie składają się one z trójkątnych ścian, lecz z prostych brył: sześcianów, kul, walców, stożków i innych. Dzięki dobrze przemyślanym i wydajnym narzędziom możemy stworzyć absolutnie dowolny kształt, nawet najbardziej skomplikowany, nie zaśmiecając edytora tysiącami punktów i krawędzi.

Zrobienie za pomocą Real'a prostej animacji nie jest zadaniem trudnym - wystarczy trochę czasu i chęci. Poniższy opis powinien ułatwić Wam tworzenie własnych dzieł z pomocą tego programu.

Na ekranie widoczne mamy trzy rzuty naszej przestrzeni,lecz nie są one w żaden sposób oznaczone, gdyż tylko od nas zależy ich wykorzystanie. Dla wygody przyjmujemy, że okno u góry, z lewej strony to widok z przodu (1), z prawej - widok z boku (2) i na dole - widok z góry (3). Obsługę edytora bardzo ułatwia nakładka aktywizująca automatycznie okno znajdujące się pod pointer'em. Dla Kickstart'u 1.3 będzie to np. program DMouse, a dla 2.0 standardowy AutoPoint. Jest to szczególnie wygodne gdy okna Real'a są w trybie Overlap. Jeśli takiego programu nie posiadamy, przed każdą zmianą aktywnego okna musimy na nie kliknąć, aby uzyskać prawidłowe współrzędne dla danego rzutu.

OBIEKTY

Nasza animacja składać się będzie z dwóch, prostopadłych do siebie luster, od których odbijać się będzie gumowa piłka. Rozpoczynamy od stworzenia luster. Z menu Colors wybieramy biały, a z Creation/Tools - Fence, lub odpowiednią ikonę. Za pomocą tego narzędzia rysujemy w oknie 1 dwa odcinki o współrzędnych (-150,50,0), (0,-100,0) i (150,50,0). Kończymy prawym guzikiem myszy - otrzymaliśmy obiekt składający się z dwóch prostopadłych prostokątów. Jeśli coś nam nie wyjdzie możemy całą operację odwołać klawiszami Shift+U. Następnie przechodzimy do okna 3 i za pomocą Modify/Hierarchy/Stretch rozciągamy dwukrotnie nasze lustra. Jako punkt względem którego rozciągamy obiekt wybieramy (0,50,100), chwytamy w punkcie (0,50,0) i przemieszczamy do (0,50,-100) - w efekcie został on powiększony dwukrotnie i wypełnia prawie całe okno. Lustra mamy już prawie gotowe - teraz kolej na piłkę.

Przechodzimy do "katalogu głównego" naszej sceny klikając na szczycie listy obiektów. Wybieramy kolor czerwony i za pomocą Creation/Primitives/Sphere (lub ikony, co jest chyba prostsze) umieszczamy w dowolnym oknie kulę o współrzędnych (0,50,-30) i promieniu 40.

MATERIAŁY

Aby lustro było lustrem musi być wykonane z odpowiedniego materiału, a zdefiniowanie takiego w Real'u jest banalnie proste. Z menu Projects/Materials wybieramy Create. Pojawia się okno, w którym możemy ustawić wszystkie parametry kreowanego materiału. W polu Name wpisujemy "mirror" i następnie ustawiamy: Brillancy (odbicie światła) 80, Transparency (przezroczystość) 0, Speed of light (załamanie światła) 98, Turbidity (mętność) 0, Specularity (połysk) 30, Spec. brightn. (jasność połysku) 15, Bump height (wysokość bump-mapping'u) 0. Resztę pozostawiamy bez zmian.

Wartości te można oczywiście zmodyfikować i trochę poeksperymentować z różnymi rodzajami powierzchni. Tworzymy także drugi materiał, z którego zrobimy piłkę. Nazywamy go "plastic" i ustawiamy: Brillancy 0, Transparency 0, Speed of light 100, Turbidity 0, Specularity 80, Spec. brightn. 30, Bump height 0. Gdy już skończymy, z listy obiektów wybieramy "obj", czyli nasze dwa lustra i za pomocą funkcji Modify/Hierarchy/Material wybieramy "mirror". Podobnie postępujemy z piłką "sphere" przypisując jej "plastic".

ŚWIATŁA

Światła mogą być oczywiście barwne, lecz my zadowolimy się zwykłymi - białymi, wybierając ten kolor z menu Color. Naciskamy klawisz "-" (może być z klawiatury numerycznej, jeżeli mamy zainstalowaną odpowiednią mapę) co spowoduje zwiększenie widocznego w oknach obszaru dwukrotnie i umieszczamy lampy, przechodząc oczywiście do głównego katalogu naszej hierarchicznej sceny.

Pierwszą z nich umieszczamy w punkcie (0,220,0), a współrzędne drugiej to (0,-60,220) - tak więc mamy oświetlenie z przodu i z góry.

Światła możemy tworzyć za pomocą funkcji Creation/Lamp lub ikony przedstawiającej słońce. Naciśnięcie klawisza "+", lub użycie funkcji Reset z menu Settings/Display przywróci początkową skalę okien.

PERSPEKTYWA

Do sterowania perspektywą służy w Real'u osobny moduł. Przechodzimy do niego wybierając Modes/Wireframe. Na ekranie otrzymujemy perspektywiczny widok naszej sceny ukazany za pomocą liniowego szkieletu. W polu Aimp ustawić możemy współrzędne punktu patrzenia (można to zrobić w edytorze za pomocą Creation/Aim point), a w polu Pos - położenie obserwatora (w edytorze - Creation/Observer). Wartości te pozostawiamy bez zmian t.j. (0,0,0) i (0,0,1000). Za pomocą suwaka Screen regulować możemy kąt patrzenia - wpisujemy tam 175, co daje perspektywę szerokokątną. Natomiast suwak Dist służy do sterowania odległością obserwatora od punktu patrzenia. Aby przybliżyć się do obiektów nadajemy mu wartość 120. Tak ustawione parametry perspektywy zatwierdzamy klikając na REC i wychodzimy do edytora.

ANIMACJA

Rozpoczynamy od ustawienia rozmiaru naszej animacji w Projects/Animation/Size - wpisujemy 25 klatek. Pomiędzy poszczególnymi klatkami możemy się teraz przemieszczać za pomocą strzałek na górze ekranu, lub funkcji Goto frame z menu Projects/Animation.

Na początek udajemy się do klatki 4, wybieramy z listy sferę i klikamy na symbol (X) znajdujący się pomiędzy wspomnianymi strzałkami. Od tej chwili sfera w klatce 4 jest "naświetlona" (ang. exposed), co oznacza, że jakiekolwiek operacje wykonane na wcześniejszych klatkach 0-3 nie będą miały na nią wpływu. Funkcję Expose możemy zlikwidować za pomocą Projects/Animation/De-expose i podając numer klatki. Wracamy teraz do klatki 0 i za pomocą Modify/Hierarchy/Size powiększamy naszą sferę 1.3 raza względem jej środka. Aby operację tę wykonać dokładnie sferę należy chwycić na przecięciu krawędzi z jedną z jej osi. Następnie w oknie 1 spłaszczamy ją także względem środka funkcją Modify/Hierarchy/Stretch - wystarczy chwycić ją w punkcie (0,102,-30) i przesunąć go do (0,80 -30). Kolejną operacją będzie obrócenie jej o -45 stopni (także względem środka) funkcją Modify/Hierarchy/Rotate i przesunięcie tak, aby dotykała lewego lustra. Wybieramy Modify/Hierarchy/Move, chwytamy piłkę w punkcie leżącym na przecięciu krawędzi z osią prostopadłą do lustra i przesuwamy do (-86,-16,-30).

Teraz skaczemy do klatki 12 i znowu korzystamy z Expose. Wracamy do 4, obracamy kulę i przesuwamy tak jak poprzednio. Z menu Projects/Animation wybieramy Morphing - klatkę początkową ustawiamy na 0, końcową na 4, a parametr Curvature pozostawiamy bez zmian, czyli 5. Aby ocenić wyniki naszej dotychczasowej pracy skorzystać możemy z Projects/Animation/Play - efekt ładny, ale widać, że to niestety jeszcze nie koniec pracy. Analogiczne operacje wykonać musimy na końcu animacji - lądowanie piłki na prawym lustrze. W tym celu "naświetlamy" sferę w klatce 24 i wykonujemy działania podobne jak na klatce 0, lecz w stosunku do prawego lustra.

Potem znowu "naświetlamy", tym razem klatkę 20 i obrabiamy ją tak jak 4. Morphing tym razem wykonujemy od 20 do 24. Nasza piłka miała jednak płynnie się odbijać, a nie przeskakiwać - to na szczęście nie jest żadnym problemem. Z pomocą przychodzi funkcja Projects/Animation/Key framing, która robi automatycznie gładkie przejścia pomiędzy "naświetlonymi" klatkami. Podajemy zakres klatek od 4 do 20, a Real dzięki algorytmom interpolacji zaanimuje naszą piłkę po krzywej przechodzącej przez środki kul w klatkach 4, 12 i 20. Kompletną już animację należy zapisać za pomocą Projects/Animation/Save, aby nasze dotychczasowe wysiłki nie poszły przypadkiem na marne.



RENDERING

Wybieramy teraz Modes/Wireframe i naciskamy gadget PLAY - oglądamy szkieletową postać naszej animacji, którą możemy zatrzymać korzystając ponownie z PLAY. Gadget'y << i >> dają nam możliwość przemieszczania się pomiędzy kolejnymi klatkami. Jeśli wszystko jest w porządku możemy przejść do ostatniej fazy naszej pracy klikając na SOLID.

W tym module możemy ustawić wszystkie parametry gotowej animacji - wielkość, rozdzielczość, sposób rendering'u i inne. W polu Mode włączamy NORMAL, a w Options pozostawiamy włączone tylko AUTOLIGHT. BASELIGHT ustawiamy na 4,4,4, Brightness 20, Overlight 1, Antialiasing 1, Resolution 1*1, Recursion depth 3. Jeśli dysponujemy tylko 1 MB pamięci i nie mamy dysku twardego, to szerokość (Width) ustalamy na 160, a wysokość (Height) na 128. W polu Name podać należy nazwę pod jaką będą zapisywane na dysku kolejne klatki animacji np. DF0:Anim. Gdybyśmy chcieli obejrzeć jedną klatkę bez zapisywania, należy ustawić jej numer w polu Frame i włączyć SINGLE. Teraz kolej na najbardziej czasochłonną część - do Frame wpisujemy 0 i naciskamy RENDER.

W czasie gdy nasz komputer będzie wyliczać kolejne obrazki możemy pójść na długi spacer, bo trochę to potrwa. Gdy już skończy, możemy wyjść z Real'a i uruchomić programik DeltaConvert. Tryb konwersji ustawiamy na Anim5, podajemy nazwę pliku, w którym zapisana zostanie gotowa animacja, następnie nazwę zapisanych uprzednio klatek (DF0:Anim) i ich zakres od 0 do 24. Nasze "arcydzieło" możemy teraz obejrzeć korzystając np. z DeluxePaint'a i funkcji ping-pong. Gdy znudzi się Wam oglądanie odbijającej się piłki może popróbujecie stworzyć coś bardziej ambitnego...

Konrad Dubiel



DATA-TOSTER

Pamiętacie jak kilka numerów KEBABA temu prezentowaliśmy napisaną w języku BASIC bazę danych? Jak się okazało - oddźwięk był spory - Czytelnicy pytali się, czy nie można by np. polskie litery, sortowanie w/g polskiego alfabetu itd. Cóż było robić - przysiadłem i naskrobałem nową bazę danych dla naszych Czytelników. Obecna wersja jest napisana całkowicie w języku maszynowym co oznacza jej szybszą pracę i posiada wbudowane definicje polskich znaków (i oczywiście je uwzględnia w podczas operacji sortowania). Myślę, że niebagatelny jest fakt zainstalowania procedur szybkiego zapisu i odczytu danych z dyskietki. Tym samym posiadacze magnetofonów zostali pokrzywdzeni, ale naprawdę ilość ustępstw, które należało by zrobić, uczyniło by używanie tego programu bezsensownym.

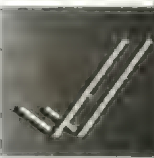
Przed rozpoczęciem jakiejkolwiek pracy z programem DATA-TOSTER należy go wprowadzić do pamięci komputera. Listing został wygenerowany w systemie INPUTER, toteż do jego wpisania należy użyć właśnie tego programu (zamieszczonego w poprzednich KEBABACH) a całość nagrać na dyskietkę.

Gotowy wpisany program jest skompresowany i każdorazowo go uruchamiając należy poczekać ok. 3 sekund na jego dekompresje. Po uruchomieniu DATA-TOSTER próbuje znaleźć na aktualnie umieszczonej w stacji dyskietce zbioru typu SEQ o znazwie "Start-Up". O ile taki zbiór istnieje i został wygenerowany przez tą bazę danych to zawiera nazwę bazy do automatycznego załadowania. W ten sposób można zaoszczędzić czas, który bez tej możliwości należałoby poświęcić na wpisanie tej nazwy "od ręki". W przypadku nie znalezienia "Start-Up" na dysku program przechodzi do menu głównego bazy. Wszystkie opisy zostały wykonane w języku polskim, co powinno znacznie ułatwić obsługę.

Podczas zakładania bazy danych program wymaga od nas ustalenia kilku rzeczy, które zostaną pokrótce omówione. Po pierwsze, należy podać nazwę bazy. Nazwa ta będzie występowała na ekranie podczas większości operacji związanych z tą bazą. Następnie komputer prosi nas o zdefiniowanie pól (kluczy) występujących w każdym rekordzie bazy. Można ich przydzielić maksymalnie 10 (1,2,3,4,5,6,7, 8,9,0). Po podaniu nazwy i wciśnięciu RETURN, pod wpisaną nazwą ukazuje się tekst "+/-" i rząd kropek. Operując klawiszami "+" i "-" można zmodyfikować długość owego rzędu, co odpowiadać będzie ilości znaków przyznanych dla zawartości danego pola (1 kropka = jeden znak). Ustalanie wielkości kończymy poprzez wciśnięcie RETURN (program automatycznie przechodzi do pytania o nazwę następnego pola). Nie musimy wykorzystywać wszystkich kluczy, bo może nam wystarczyć np. 5 pól. Aby tak właśnie ustalić tą ilość, wystarczy nie podawać nazwy dla pola nr 6, poprzestając tylko na wciśnięciu RETURN. Następnie wprowadzić należy nazwę, pod jaką zbiór z danymi będzie nagrywany na dysku. O ile np. wprowadzimy nazwę "książki", to na dyskietce uzyskamy zbiór o nazwie "b.książki". Literka "b" oraz kropka są dodawane automatycznie i służą programowi do rozpoznawania swoich zbiorów.

Po wprowadzeniu danych dotyczących naszej bazy, program domaga się wprowadzenia choćby jednego zapisu. Po jego wpisaniu przechodzi do opcji operacji na danych. Na ekranie operacyjnym wydrukowane są takie dane jak nazwa bazy, ilość zajętych rekordów, ilość możliwych do zapisania rekordów (rekord - z j. ang. - zapis), dane aktualnego rekordu a pod spodem spis komend możliwych do wywołania. Nazwy komend są oddzielone za pomocą przecinków, a ich wywołanie odbywa się poprzez wciśnięcie litery sterującej, która w nazwie każdej z funkcji jest wydrukowana jako litera duża. Część operacji dotyczy załych rekordów (np. kasowanie lub dopisanie nowego rekordu) i w takim przypadku działania odnoszą się do wszystkich pól w danym rekordzie. W przypadku działań, w których argumentem jest jedno z pól (np. sortowanie, wyszukiwanie czy modyfikacja) pole takie powinno być uprzednio wskazane poprzez ustawienie przed nim strzałki kontrolnej (jest stale widoczna przy lewym marginesie ekranu). Ustalanie jej pionowego położenia odbywać się powinno poprzez klawisze ruchu kursorem. Umieszczona w programie opcja sortowania (bąbelkowego) sortuje dane w/g zapisów we wskazanym polu posługując się kolejnością liter w alfabecie polskim.







Powrót z opcji operacji na danych do menu głównego powoduje zniszczenie wszystkich danych w bazie. Aby zapobiec niekontrolowanemu zniszczeniu danych, po dokonaniu jakichkolwiek modyfikacji ustawiany jest specjalny znacznik w programie. Jego zadaniem jest spowodowanie upomnienia się przez program o sporządzenie zapisu na dysku (odpowiada się T [tak] lub N [nie]). Warto zauważyć, że ostatnia zapisana na dyskietce wersja zostaje przemianowana na "!.nazwa bazy", co funkcjonalnie stanowi odpowiednik zbiorów BAK na np. komputerach klasy PC. W przypadku nagrania błednych danych, poprzednie zawsze można odzyskać poprzez wybranie opcji "poprzednie dane" w czasie operacji na zapisach. Staremu zbiorowi przywracana jest wówczas nazwa widziana przez program jako nazwa bazy danych ("b.nazwa bazy").

Aby uniknąc kłopotliwego (bo czasochłonnego) wywoływania żądanej bazy z menu głównego, wystarczy wpisać jej nazwę do zbioru "start-up", przy pomocy jednej z opcji menu głównego. Jak już wspomniano na początku tego opisu - tak wskazana baza jest potem automatycznie wgrywana podczas uruchamiania samego DATA-TOSTER'a. U mnie program znakomicie pracuje jako podręczna książka adresowo-telefoniczna. Mam nadzieję, że przyda się również i w Waszych domach.

Paweł "POLONUS" Sołtysiński

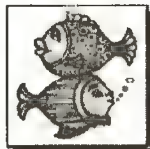
HOROSKOP 1993

Wodnik
21.I - 20.II



W nowym roku czeka Cię wiele niespodzianek: cena komputera, którego zakup planowałeś na ten rok, podskoczy do góry tak dalece, że nie pomoże nawet wyprzedaż rodzinnego majątku. Jeżeli planowałeś zmianę grupy, w której aktualnie się udzielasz jako koder czy grafik, na nową prestiżową grupę, to możesz być pewien realizacji pierwszego etapu tego przedsięwzięcia, tzn. zostaniesz wywalony z hukiem ze starej. Koniec roku będzie jednak pomyślniejszy, gdyż błyszczeć będziesz w glorii chwały jako wyróżniony w plebiscycie lokalnego magazynu komputerowego "Lamer Roku" ew. zdobywca "Złotej Klamki" (np. w KEBABIE).

Ryby
21.II - 20.III



Więcej wiary w siebie! Powinieneś pamiętać o tej radzie zwłaszcza w wielu czekających Cię w nowym roku momentach ciemnoty umysłowej podczas pracy z Twoim komputerem. Dobrze znasz to uczucie - z ekranu mruga do Ciebie niecierpliwie kursorkami jakaś tam MasterSeka lub TurboAssembler a Ty siedzisz jak ten nie przymierzając Dr Boczek z beznadzieją w oczach. Ale nie martw się, są upadki i wzloty a Ty przecież upadasz jak kot na cztery łapy.

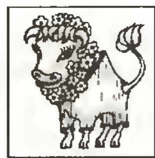
A jak Ci nie wyjdzie zabawa z komputerem, zawsze możesz wrócić do zabawy z kostką Rubika; jeżeli i to będzie za trudne to proponuję spróbować układać Puzzle 1000-elementowe. Dwustronną listę następnych w kolejności substytutów zamieścimy w następnym numerze KEBABA.

Baran
21.III - 20.IV



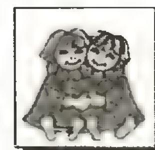
Pamiętaj, że komputer to nie worek treningowy i walenie w niego pięścią tudzież przemawianie w sposób dobitny nie usunie błędów z pisanego przez Ciebie programu. Ale metodę gorąco polecam - opłata za naprawę komputera skutecznie usuwa błędy ze sposobu naszego rozumowania. W okresie wakacyjnym spędzisz miło czas na kilku Copy-Party, na których ukradną Ci wiadro płaskokrążków informacjonośnych, pourywają kable i zapuszczą wirusa na Twoje dyski, a Ty zaliczysz kilka noclegów na dworcach w nieznanych miejscowościach (tak to jest, jak się wsiada do pociągu po pijanemu).

Byk
21.IV - 21.V



Jeżeli tradycyjnie masz kłopoty ze zdecydowaniem, jaki to komputer sprawić sobie w nadchodzącym roku - nie spiesz się! Planowałeś zakup Amigi 4000 - odpuść sobie. Znając pokrętną politykę rynkową firmy Commodore, na Boże Narodzenie 1993 spodziewać się będzie można np. Amigi 10000, która będzie totalnie niekompatybilna w dół, wyposażona w nowe, zaskakujące gniazda przyłączeniowe (tym razem podobne do porcelanowych wtyczek używanych przy prodiżach naszych Mam) i poprawionym Interlace, od którego kineskop wypadnie Ci na biurko. A w ogóle to się zabierz za naukę, bo nowymi High-Score'ami to Ty kolejnej klasy nie zaliczysz...

Bliźnięta
22.V - 21.VI



Osoby spod Twojego znaku zwykle dobrze znajdują się w różnych sytuacjach, które wymagają zapału i zdolności organizacyjnych. Wielu z Was to redaktorzy naczelni dyskietkowych magazynów komputerowych, które kończą swój żywot po pierwszym numerze. Spróbuj, możesz być odstępstwem od tej reguły! Udowodnij ludziom ze swojej grupy jak bardzo jesteś przydatny i zasypuj ich kilogramami procedur (jeżeli jesteś koderem), toną nowych obrazków (jeżeli jesteś grafikiem), zbiorami atonalnych dźwięków (jeżeli nie jesteś muzykiem) lub puszkami z Coca Colą, piwem, hamburgerami i biletami do kina (jeżeli jesteś sponsorem grupy, tzn. osobą, która ma pieniądze, ambicje i to by było na tyle).

Rak
22.VI - 22.VII



Postaraj się zaplanować ten rok dokładnie, gdyż nadgonienie wszystkich nowinek programowych obecnych na scenie (a których brakuje w Twoich programach demonstracyjnych) zajmie Ci trochę czasu. Jeżeli Ci się to nie uda, to zawsze możesz oświadczyć, że z Commodore 64 przechodzisz na Amigę "bo na niej to dopiero można". Jeżeli nie będzie Ci się wiodło na Amidze, zawsze możesz kupić jej nowszą wersję np. Amigę 4000. Jeżeli i ta rozpaczliwa zmiana nie przyniesie rezultatu - zawsze możesz wszystko sprzedać i powiedzieć, że nie masz czasu na głupoty.

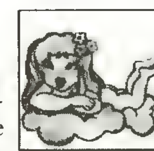
Lew
23.VII - 22.VIII



Nowy rok przyniesie Ci wiele wciągających gier. Będziesz zjadał, strzelał i udeptywał wszystko, co będzie się panoszyć po ekranie Twojego komputera. Przypuszczalnie prawa autorskie wciąż nie będą obowiązywać; ciężko jest bowiem znaleźć czas na uchwalenie odpowiednich prawnych konsekwencji do czasu, kiedy nasi posłowie nie znajdą skutecznego sposobu na wymuszenie wyżu demograficznego w naszym kraju.

Dobra rada na ten rok: wciągnij Twoją rodzinę do wspólnej zabawy z komputerem. Zwykle są dwa efekty takiego postępowania - a) Twoi bliscy uświadomią sobie naocznie, jak bardzo nie są w stanie pojąć wszystkiego, co związane jest z komputerami i dają Ci spokój lub b) zapalają się do wspólnej zabawy i wykazują znacznie więcej zrozumienia dla Twoich zainteresowań ,a stąd już tylko jeden krok do rodzinnych zawodów w np. Panza Kick Boxing.

Panna
23.VII - 22.IX



Jesteś wprawdzie urodzonym pedantem ale pamiętaj o zachowaniu umiaru w ten sposób, by nie poświęcać proporcjonalnie większej ilości czasu na śliczne opisywanie posiadanych dyskietek niż na samodzielne próby programowania. Jest to jednak dobra cecha dla komputerowych twórców grafiki, więc jeżeli jeszcze tego nie próbowałeś - spróbuj, a być może szybko wzbijesz się na wysokie miejsca w różnego rodzaju "Listach Najlepszych", czego Ci z całego serca życzę. Szczęście w tym

roku przynosić Ci będzie regularne kupowanie KEBABA.

Waga
23.IX - 23.X



Ten rok upłynie Ci pod hasłem dokupywania potrzebnego sprzętu. Posiadacze magnetofonów staną się posiadaczami magnetofonów dwukasetowych, użytkownicy drukarek dokupią do nich jedną igłę więcej, a Ci, którzy na co dzień pracują przy użyciu myszki - dorobią do niej auto-fire. Osobom rozglądającym się za gustownym mebelkiem pod komputer polecamy wyszukanie w sklepie meblowym czegoś z długim blatem (przecież trzeba to wszystko gdzieś postawić) i miejscem na wyciągnięcie nóg. Sugestie, jakoby najlepiej spełniała te warunki trumna, nie są najrozsądniejsze - wydajność wentylacji przedstawia wiele do życzenia a poza tym ktoś mógłby wziąć napis "GAME OVER" zbyt dosłownie...

Skorpion
24.X - 22.XI



Osobom spod tego znaku w nadchodzących miesiącach powinny się powieść wszystkie zamierzenia związane z ich komputerową częścią życia. Jako jednostki z natury wrażliwe i ambitne starać się będą o wnoszenie ciągłych twórczych urozmaiceń. Osoby tworzące własne programy powinny liczyć na przypływ pomysłów raczej lepszych niż użycie po raz kolejny jakiegoś Demo Maker'a. Jeżeli nadal będziesz mieć do tego ostatniego ciągoty, to oznacza to, iż niechybnie tak naprawdę musisz być spod innego znaku (bo np. zostałeś zamieniony przez nieuwagę jeszcze w szpitalu) lub nie czytujesz tych horoskopów, co trzeba. Niesamowite, jak można żyć w takiej nieświadomości.

Strzelec
23.XI - 21.XII



Strzelec jest razem z Baranem i Lwem w tzw. trygonie ognistym (określenie wbrew pozorom nie wzięło się od wody ognistej), co oznacza, że pomiędzy osobami spod tych znaków jest szczególnie łatwo o porozumienie. Wypływa z tego nauka o tym, że o ile jest to możliwe, należy dobierać osoby do grupy komputerowej w/g znaków Zodiaku.

Należy jednak z tym nie przesadzać, bo np. grupa złożona z samych baranów... W roku 1993 możesz jednak się spodziewać wielu nie zapomnianych chwil spędzonych z Twoim komputerem, nowych złośliwych wirusów i przesyłek skuteczniej wygniecionych przez pocztę.

Koziorożec
22.XII - 20.I

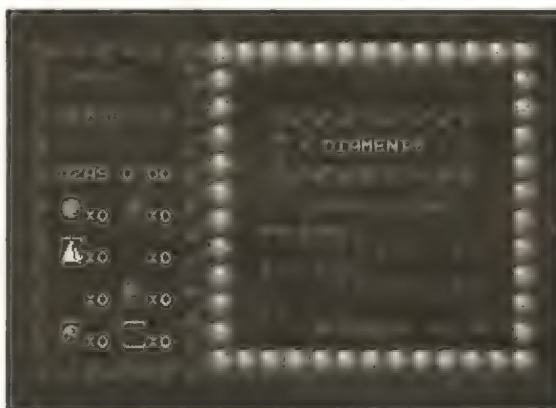


W tym roku będzie nad Tobą świecić szczęśliwa gwiazda i o ile wykorzystasz jej wspomagające pracę po nocach światło, zdarzyć się może, że uda Ci się osiągnąć niedostępne dla innych techniki programowania Twojego komputera. Tak więc przestań myśleć o głupotach, dziewczynę zostaw w spokoju, ucz się tak aby Cię tylko na zbitą klawiaturę ze szkoły nie wyrzucili.

Cały pozostały czas skrupulatnie poświęcaj na pracę z komputerem (nie zapomnij zawczasu umówić wizyty u okulisty, gdzieś tak po wakacjach). A propos wakacji - nie warto kupować olejku do opalania bo i tak nie będziesz miał czasu na kąpiele słoneczne. I bez nich udar mózgu zapewniony.

DIAMENTY

Znana zapewne naszym Czytelnikom z poprzednich numerów firma *SONIX* wydaje się specjalizować w produkcji prostych gier logicznych. Mieliśmy okazję obejrzenia kolejnej produkcji - gry zatytułowanej **Diamenty**. Ten sam programista, podobny wygląd zewnętrzny i podobny schemat - "popełniłeś błąd, więc zaczynaj wszystko od nowa".



Polem gry jest ograniczony "kamieniami" obszar (lub obszary), we wnętrzu którego znajdują się tytułowe diamenty. Wbrew oczywistym oczekiwaniom, ich zbieranie nie odbywa się w żaden zwykły sposób - należy się ich bowiem ...pozbywać!

Przypomina to trochę przekraczanie "masy krytycznej", gdyż doprowadzenie dwóch lub więcej diamentów tym samym kolorze (kształcie) do zetknięcia powoduje ich dosłowną dematerializację.

Jak nietrudno zauważyć, do skutecznego zniknięcia potrzebne są przynajmniej dwa diamenty (należy więc uważać, by nie zostać z jakimś pojedynczym diamentem na końcu...).

Nie jest to takie proste, zważywszy na fakt, że znajdujące się "w pokładzie" diamenty podlegają siłom ciążenia i wobec tego raczej nie należy liczyć na to, że zawieszony bez podparcia diament będzie miał ochotę tak sobie wisieć.

Gra mimo wszystko nie jest nużąca i dla miłośników gier logicznych (zwłaszcza tych od kostki Rubika) powinna stanowić pewnego rodzaju wyzwanie (nie jest łatwo ją ukończyć).

Mamy nadzieję, że Autor (Zenon Mikołajczyk) w przyszłości będzie potrafił zaskoczyć nas świeżością rozwiązań, co pomóc powinno jego produkcjom wyglądać jeszcze bardziej interesująco.

Redakcja.

C-64 GAMES

DEMON BLUES (c) MICROVALUE

Jedna z ciekawszych labiryntówek, jakie pojawiły się ostatnio. Mimo, że nie zajmuje całej strony dysku (istnieje w wersji kasetowej), to oferuje nam kolorową i ciekawą grafikę. Sama gra przypomina troche dość stary już BLINKY'S SCARY SCHOOL, ale jest trochę różnic(głównie w zadaniu, które mamy do spełnienia). Na pewno wyróżnia sie z pośród wielu labiryntówek, ktore się ostatnio ukazały.

SCENARIO (c) STARBYTE SOFTWARE

Dość ciekawa (koneserzy twierdzą, że bardzo dobra!) gra strategiczno-zręcznościowa. Wcielamy się w postać władcy jednego z państw fikcyjnej Europy i naszym zadaniem jest osiągnięcie dominacji nad pozostałymi graczami. Ciekawą opcją jest możliwość grania kilkoma graczami na raz. Grafika scenek zręcznościowych pozostawia trochę do życzenia.

ELVIRA II (c) FLAIR SOFTWARE

Rewelacyjna kontynuacja ELVIRY. Oszałamiająca grafika, jeszcze bardziej rozbudowana fabuła (całość zajmuje 4 dyski!) i dużo bardzo dobrej zabawy. Oto jak w skrócie można opisać najnowszy produkt firmy FLAIR SOFTWARE. Gorąco polecamy wszystkim miłośnikom gier przygodowych.

ET'S RUGBY LEAGUE (c) AUDIOGENIC

Świetna symulacja gry w rugby. Sposobem prezentacji graczy i pola gry przypomina Kick off'a z Amigi, a sama gra jest na pewno przyjemniejsza niż na wszystkich symulatorach piłki nożnej na C64 (oczywiście musimy najpierw opanować zasady rugby...). Program pozwala rozgrywać akcje na wiele różnorodnych sposobów a przeciwnik gra dosyć inteligentnie...

GREYSTORM (c) DOUBLE DENSITY

Wszyscy zapewne pamiętamy rewelacyjnego HAWKEYE. Ta gra jest oparta na dokładnie takim samym pomyśle i zawiera podobne rozwiązania graficzne (dwa plany animacji itp.). Grafika jest bardzo dobrze dopracowana (dwa plany robią wrażenie...). Całości towarzyszy bardzo dobra muzyka autorstwa 20CC. Dużo dobrej zabawy i frajda dla miłośników gier zręcznościowych.



STONE AGE (c) ECLIPSE

Kolejna (z wielu ukazujących się ostatnio) gier logicznych. Od swoich poprzedników różni się przede wszystkim doskonałą grafiką (szczególnie w intrze i w menu) oraz dużą ilością muzyki (m.in. dziesięć utworów do wyboru jako podkład w czasie gry). Sama gra polega na przeprowadzeniu myszki do wyjścia poprzez szereg klocków. Część z nich jest ruchoma (możemy przesuwać je kursorem) a pozostałe też mają różne funkcje. Sens gry jest zbliżony do LEMMING'ÓW. Polecam.

HOOK (c) OCEAN

Kolejna gra firmy Ocean zrobiona na podstawie filmu. Tym razem na podstawie znanego dzieła Stevena Spielberga. Wersja C64 różni sie od wersji na Amigę i PC. Nie jest to gra typu adventure lecz zwyczajna zręcznościówka. Niestety grafika nie jest najmocniejszą stroną tej gry. Podobnie jest z muzyką. Wprawda gra może się podobać, gdyż poziomy są naprawdę różnorodne, ale wyraźnie widać niedopracowane szczegóły.

SLICKS (c) CODEMASTERS

Wiele już napisano rajdów samochodowych dla C64. Ten należy do tych lepszych. Koncepcja gry jest zbliżona do HOT ROD, ale samo prowadzenie samochodu jest o wiele bardziej dynamiczne (podobne do INDY HEAT). Sporo tras, możliwość rywalizacji z komputerem, wiele "stajni" wyścigowych (możemy zmieniać kluby w zależności od osiąganych wyników) i niezła muzyka powodują, że w tę grę grać można bardzo długo (bardzo wciąga!). Polecam.



ENFORCER (c) GOLDEN DISC

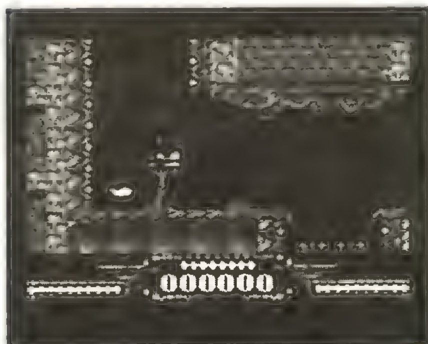
Niemałą sensacje wywołało ukazanie sie ostatnio gry ENFORCER autorstwa Manfreda Trenza. Wszyscy zapewne pamiętamy świetne gry: Turrican 1 i 2, Katakis czy R-type. Po ukazaniu się Turrican'a autor zapowiedział, że definitywnie rozstaje się z komciem. Jak widać swojej obietnicy nie dotrzymał (i słusznie!) a w planach ma jeszcze wiele nowych gier. Parę słów o samej grze. Przez graczy jest często nazywana KATAKIS 2 co powinno wiele powiedzieć na temat jej treści. Doskonała strzelanka ze świetną grafiką i bardzo dużą ilością obiektów latających. Szczególnie duże wrażenie robią finałowi przeciwnicy na końcu każdego poziomu. Gra jest dość trudna, ale sześć poziomów z różnorodną grafiką gwarantuje długą i dobrą zabawę.

COOL WORD (c) OCEAN

Tym razem Ocean zdziwił nas wszystkich wyjątkowo dobrą muzyką i grafiką.

Dość sympatyczna labiryntówka zrobiona (jak większość gier Ocean'u) na podstawie filmu. Osiem dość długich poziomów i szybka animacja, oto cechy dzięki którym gra ta może się podobać.

Krzysztof "BRUSH" Dąbrowski







OGŁOSZENIA DROBNE

Poszukuję programów: Master Scenery Disc, California Challenge Scenery Disc, European Challenge Scenery Disc, LEMINGS (C-64)
Marcin Porankiewicz
ul. Konopnicka 23/2
72-100 Goleniów

Sprzedam C-64, 1541 II, FINAL III, 100 dyskow, 20 kaset, magnetofon, monitor B-W, 2 joysticki, literatura, mysz, pokrywa.
Grzegorz Miodek
ul. K. Wielkiego 20/8
47-220 Kedierzyn Koźle

Sprzedam C-64, magnetofon, drukarkę D1[illegible]M + interfejs, X, Final III, 27 kaset, literatura. Cena 3 mln.

Szymon Grygowski tel. 20-481
ul. Poniatowskiego 22/27
72-200 Nowogard

TANIO sprzedam A500+, b. dobre rozszerzenie pamięci Gigatron 1Mb, tel. 536-235, po 15.00
Grzegorz Grabek
ul. Wiosny Ludow 1/4
71-471 Szczecin

Gry, użytki, dema na C-64. Katalog. Koperta + znaczek
Tomasz Tra[illegible]owski
ul. Nawojowa 21/[illegible]
3[illegible]-300 Nowy Sącz

Sprzedam C-64, magnetofon, FINAL II, Ex-Plus, joystick, 12 kaset, 6 książek, zapasowy zasilacz. Cena: 1,9 mln.

Mirosław Kajstura
ul. Jaśminowa 17
43-410 Kończyce

Sprzedam: C-128 D, Neptun 156, magnetofon, 2 joysticki, ACTION plus 6.0, 80 dysków + box, literatura. Cena 6mln.
Michał Czajkowski tel. [illegible]8-573
ul. Komuny Paryskiej [illegible]/34
8[illegible]-858 Bydgoszcz

Amiga. Klub korespondencyjny. Gry i użytki darmo. Szczegóły: koperta zwrotna + znaczek (dopisek: Amiga)
Marcin Mirkowski
ul. Maltanska 1/93
02-761 Warszawa

Skontaktuję się z posiadaczem gry Karateka na C-64. W zamian inne gry na taśmie.
Tomasz Pająk
ul. Kasztanowa [illegible]
Wał[illegible]rzych

Kupię mysz [illegible] (z dyskietką) do C-64. Podać opis i cenę.
Karol Skowroński
ul. Kościuszki 17/6
12-100 Szczytno

Nawlążę kontakt z grającymi w "FLIMBO'S QUEST"
Michał Orłowski
ul. Żeromskiego 2B/5
73-110 Stargard

Doktor Boczek donosi...

Jednym z alternatywnych sposobów na wyeliminowanie dokuczliwego migania obrazu w trybie interlace jest zastosowanie nowej przystawki o nazwie Chair 2000. Skomplikowany system siłowników sterowanych przez komputer wprowadza krzesło na którym siedzi użytkownik w zawiłą sinusoidę drgań, zsynchronizowanych z częstotliwością ramki. Rezultatem jest obraz o jakości nie ustępującej najlepszym, konwencjonalnym Flicker-Fixerom.

Użytkowanie tego typu przystawki wiąże się jednak z pewnymi niedogodnościami - wysoce niewskazane jest spożywanie pokarmów w czasie pracy z urządzeniem. Również osoby cierpiące na chorobę morską powinny unikać dłuższych kontaktów z Chair 2000. Pewnym rozwiązaniem wydają się tu być tabletki Aviomarin 3.0 wliczane przez niektórych dystrybutorów w cenę zestawu.

Każdy użykownik Amigi marzy o tak zwanym dopalaczu, czyli karcie przyspieszającej jej działanie. Karty te z zamontowanym procesorem przeważnie 68030 i 32-bitową pamięcią, są dosyć drogie i przeważnie przygotowane do współpracy z Amigami 2000 i nowszymi. Otóż członkowie grupy The Smilents odkryli nowy sposób na przyspieszanie działania programów. Skorzystali oni z pewnej znanej, ale dotychczas niewłaściwie wykorzystywanej właściwości Amigi. Nasz komputer wyświetla ekran 50 razy na sekundę (częstotliwość systemu PAL to 50 Hz) co jest zajęciem dość czasochłonnym. Wszystkie operacje dokonywane są w przerwie w wyświetlaniu, w okresie tzw. Vertical Blank.

Jeżeli jednak obliczenia wykonywane są z pominięciem obsługi monitora, szybkość ich wzrasta około pięćdziesiąt razy, a w systemie NTSC sześćdziesiąt (częstotliwość systemu PAL to 60 Hz). Jedynym problemem jest "powiadomienie" programu, iż ma pracować w systemie Turbo. Wspomniana już grupa zmontowała wyłącznik monitora, łączący port Serial Amigi z monitorem.

Po jego wyłączeniu, Copper nie traci czasu na sterowanie wiązką elektronów, a pomaga Motoroli w dokonywaniu obliczeń zmiennoprzecinkowych, stając się swego rodzaju koprocesorem. Oprócz samego wyłącznika, w przystawce znajduje się maleńki pojemniczek przechowujący bity kontrolne systemu Turbo. Właśnie dzięki tym bitom program dowiaduje się o nowych możliwościach płynących ze zwiększonej szybkości. Rozwiązanie takie nadaje się jedynie do zastosowań nie wymagających monitora np. "liczenie" obrazków w raytracingu. Koszt przystawki to ok. 0,5 DM + instrukcja (tom 1,2,3) 998 DM. Producent: The Smilents Inc.

Niedługo w Nowym Yorku odbędzie się proces sądowy, którego przebieg nie powinien ujść uwadze żadnego Amigowca. Otóż znany koder grupy Red Sector - Delta, oskarżył producentów filmu Terminator II, o bezprawne wykorzystanie do robienia efektów specjalnych jego nowego, jeszcze niedostępnego w sprzedaży RSI Demomaker'a 3.0.

W Anglii powstał ostatnio Klub Miłośników X-Copy. Do jego działalności zalicza się organizowanie comiesięcznych Copy Party, gdzie następuje wymiana najnowszych wersji tegoż kopiera, organizowanie konkursów w mistrzostwie posługiwania się nim i wydawanie miesięcznika "X-Copy User". W gazecie tej na stu barwnych stronach oprócz kursów obsługi X-Copy, chartsów (można głosować jedynie na niego, podając numer wersji), najciekawsza jest stała rybryka "Trips and Tics with X-Copy".

W celu zmiejszenia kosztów produkcji, nowa Amiga 600+ będzie wyposażona w gumową klawiaturę, procesor Z80A a jako pamięć zewnętrzną zastosowany zostanie magnetofon szpulowy (adapter Bambino w wersji 'economic').

Program Niespodzianka

Czy ktokolwiek z Was jest w stanie, oczywiście bez włączania komputera, przewidzieć co tak naprawdę robi ten, zamieszczony poniżej, krótki programik? Jesli Twoja odpowiedz jest twierdząca, to bez wątpienia zasłużyłeś sobie, Drogi Czytelniku, na naszą Nagrodę Specjalną (uścisk dłoni Naczelnego), którą to niezwłocznie prześlemy pocztą.

Zaś w przeciwnym przypadku pozostaje Ci tylko "wklepanie" poniższego prgramu i obejrzenie efektów jego działania. Dodam tylko, że nie jest Wam do tego potrzebny sprawny żaden monitor ani telewizor!

```
**************************
* Program Niespodzianka *
*                        *
* by Mr.Soft / W.F.M.H. *
**************************

Begin:
    move.b  #$00,$bfe801
    move.b  #$0f,$bfec01
Loop:
    bset    #$6,$bfee01
    bclr    #$6,$bfee01
    btst    #$6,$bfe001
    beq.s   Out
    bset    #$6,$bfee01
    bclr    #$6,$bfee01
    bra.s   Loop
Out:
    rts
```

Despotyzacja

Bóg widzi że, Sid Meier nie jest jednym z najładniejszych facetów pod słońcem,na szczęście jednak przy tworzeniu sygnowanych swoim nazwiskiem gier nie używa buźki, a wyłącznie intelektu. Jest to bez wątpienia krok słuszny, krok, który zadecydował o sukcesie gry RAILROAD TYCOON i który przyczynił się do powstania CYWILIZACJI - nie mającej chyba równych pod względem zaimplementowanego mnóstwa wzajemnych współzależności elementów.

CYWILIZACJA ma zapewne swoje źródła w o wiele starszej MEGALOMANII, gdzie jedynym sposobem odniesienia zwycięstwa nad przeciwnikami było wymyślenie leoszej od nich broni. W wypadku produktu Sida Meiera zadaniem gracza jest nie tylko walka - należy także dbać o odpowiednie tempo rozwoju społecznego i ewolucję miast. CYWILIZACJA jest niezłym wstępem do nauki myślenia wielopoziomowego.

Pomysł nie jest ani nowy, ani oryginalny. Zostajesz przywódcą pewnej wspólnoty plemiennej około 4000 lat przed narodzeniem Chrystusa. Możesz określić na jakim poziomie trudności chcesz grać, przeciwko ilu przeciwnikom oraz jaką cywilizacją (osiągalne są m.in. amerykańska, aztecka itp.).

Każda z niech posiada jakieś własne zdolności i sekrety, które bardziej lub mniej przydają się podczas późniejszej gry. Mapa zostaje wybrana losowo (opcjonalnie możesz grać na mapie Ziemi, niezbyt dokładnej, ale zawsze..), a teren jeszcze nie odkryty jest niewidoczny. Zmusza Cię to do podejmowania wypraw na ślepo (też możesz poczuć się Kolumbem).

Rozpoczynasz - zależnie od poziomu trudności - z jednym lub dwoma oddziałami osadników (SETTLERS). Mogą oni budować drogi, irygować teren i stawiać miasta. W miarę rozwoju technicznego dojdą umiejętności stawiania fortec, mostów i linii kolejowych. Najpierw jednak powinieneś postawić miasto - pierwsze miasto staje się automatycznie stolicą i tam buduje się pałac. Stolicę możesz przenieść stawiając pałac w innym mieście. Samo miasto możesz rozwijać budując w nim spichlerz (GRANARY),świątynię (TEMPLE) i masę innych rzeczy. W większości musisz płacić za utrzymanie budynków, aby więc zapewnić sobie odpowiedni przypływ gotówki należy ustalić podatki. Na rozsądnym poziomie jednak, gdyż wzrost podatków idzie w parze ze zwolnieniem postępu (niewiele funduszy przeznaczanych jest na naukę).

Oddziały wojskowe dzielą się zazwyczaj na dwie kategorie : do obrony i ataku. Oddziały do obrony charakteryzują się współczynnikiem siły w obronie (druga liczba) wyższym niż w ataku (pierwsza liczba) - np. oszczepnicy PHALANX . Oddziały do ataku natomiast mają współczynnik siły w ataku wyższy niż w obronie - np. rydwan CHARIOT .



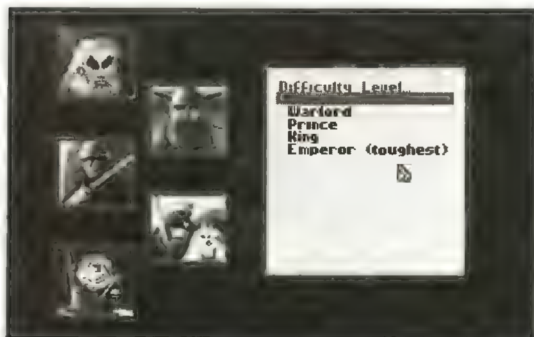
Nie znaczy to jednak, że nie możesz atakować oddziałami obronnymi czy bronić się atakującymi. Zazwyczaj jednak jest to mało efektywne i kończy się ich utratą. Pierwszym budowanym oddziałem wojska jest milicja (niezupełnie obywatelska), która jest dość słaba. W miarę odkrywania kolejnych wynalazków powinieneś budować coraz lepsze oddziały do walk na lądzie, morzu i w powietrzu, co z pewnością okaże się pomocne w zwalczaniu przeciwników. Możesz je równiesz wykorzystywać w celu tłumienia buntów w Twoich miastach ("najeżdzając" jednostką na miasto i wydając jej polecenie SENTRY, co zazwyczaj skutkuje). Przy prowadzeniu obrony ważną sprawą jest dobór pola, gdyż często nawet bardzo słaby oddział dobrze broni się np. w górach (a jeszcze lepiej w górskiej fortecy).

Pamiętaj też, że kilka jednostek postawionych na tym samym polu NIE sumuje wartości swoich współczynników obrony. W razie ataku i porażki jednego z nich - zniszczeniu ulegają wszystkie (co potrafi być denerwujące). Opłaca się rownież polować na przywódców barbarzyńców i partyzantów od czasu do czasu pojawiających się na mapie, bowiem każdy z nich wart jest stówę.

Oczywiście pozostałe cywilizacje zwykle nie nawiązują kontaktów przy pomocy mordobicia. Przeważnie wysyłają posłów z ofertą wymiany wynalazków, określenia terytoriów wpływów lub ordynarnie żądają pieniędzy. Zaraz potem deklarują czy podpiszą z Tobą pakt o nieagresji czy chcą wojny. Niekiedy przy ofercie pokoju udaje się wyłudzić od nich nieco forsy. Pakty możesz śmiało łamać jeśli czujesz się silny, ale uważaj - nie masz na to monopolu (Zulusi !).

Jeżeli nie chcesz walczyć sam, możesz zapłacić którejś z cywilizacji, aby walczyła wraz z Tobą - to jednak droga usługa. Dodatkowo, kiedy uda Ci się wynaleźć piśmiennictwo (WRITING) możesz zbudować dyplomatę (DIPLOMAT), który poza nawiązywaniem kontaktów i zakładaniem ambasad zdolny jest także do kradzieży wynalazków z miasta wroga, wzniecania w nim rozruchów, sabotowania produkcji i przekupywania wrogich oddziałów.

Osobnym tematem są Cudy Świata. Jak wiadomo w starożytności było ich siedem - w cywilizacji jest ich więcej, dokładnie 19. Wybudowanie każdego z nich daje inny efekt. Niektóre potęgują efekt innych, niektóre są cudami samymi w sobie. Zawsze jednak wybudowanie Cudu powoduje wzrost zadowolenia mieszkańców miasta, w którym się on znajduje (niekiedy we wszystkich miastach). Cudy mają oczywiście swoje ograniczenia i większość z nich jest aktywna tylko

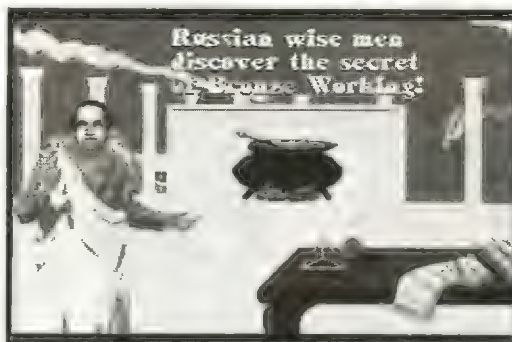


do odkrycia pewnych wynalazków. Niemniej są napewno pomocne i liczą się w punktacji końcowej. Na końcu artykułu pozwoliłem sobie dodać ich spis wraz z krótkimi objaśnieniami (spragnionych szczegółów odsyłam do CYWILOPEDII).

Nawiązując do tytułu tego artykułu - nie wziął się on z powietrza. Przy implementowaniu gier symulujących rozwój gospodarczy i społeczny daje się zauważyć swojego rodzaju syndrom despotyzmu. Doskonale zapewne znana czytelnikom UTOPIA wymuszała stawianie dużych ilości posterunków policji (aby utrzymać w ryzach przestępczość i nastroje społeczne), co niejednokrotnie doprowadzało do groteskowych sytuacji, kiedy połowa populacji pracowała w siłach bezpieczeństwa. Podobnie w REALMS, gdzie z kolei należało cały czas trzymać rękę na pulsie i interweniować w wypadku buntu któregokolwiek z podległych miast zgromadzonymi wcześniej wojskami (stały nadzór wojskowy). Niemal identycznie sytuacja ma się w CYWILIZACJI.

Trzeba trzymać wszystko żelazną ręką tyrana. W DESPOTYŹMIE istnieją wprawdzie pewne niedogodności: słabiej rozwija się handel, jest mniejszy przyrost naturalny i zyski z pól zirygowanych, ale to wszystko jest niczym ważnym bowiem nie trzeba utrzymywać wszystkich oddziałów wojskowych, a bunty w miastach zdarzają się dość rzadko i łatwo je uśmierzyć. Natomiast w REPUBLICE i DEMOKRACJI każdy oddział musi być utrzymywany z zasobów miasta, które go stworzyło, obywatele co chwilę wysuwają nowe żądania - niewykonalne z powodów finansowych, buntują się na wyścigi i na dodatek senat nie wyraża zgody na złamanie paktu z niewygodnym sojusznikiem. I jak tu wierzyć w wyższość demokracji ?

CYWILIZACJA zamieni każdego liberała w zajadłego despotę. Pod względem prezentowanej orientacji społecznej jest na poziomie zwykłej strzelaniny - dopaść i zdeptać. Amen !



Zasadniczo są trzy sposoby ukończenia gry. Pierwszy - kiedy uda Ci się militarnie spławić pozostałe cywilizacje, tak, że pozostaną po nich jedynie kawałki interesujące wyłącznie archeologów. Drugi - kiedy uda Ci się wybudować i wysłać do układu Alfa Centauri statek kosmiczny. Gdy przybędzie on na miejsce automatycznie zostajesz ogłoszony Wielkim Szefem. Jest też oczywiście sposób trzeci, naturalną koleją rzeczy wynikający z pierwszego - kiedy to Ciebie zrównają z gruntem. Oglądasz wtedy rysunek przedstawiający ile z Ciebie zostało i kto Cię odkopał.

Na zakończenie jeszcze mała niespodzianka dla amatorów "poprawiania" swojej sytuacji taktycznej za pomocą hackerskich sztuczek. W tym celu należy przy pomocy dowolnego monitora załadować plik ze stanem gry (z reguły do tego celu służy komenda '[' np.: [NazwaPliku adres) pod pewien adres bazowy (np. $30000). W okolicach adresu baza+$140 (w naszym przypadku $30140) przechowywana jest informacja o posiadanej przez nas kwocie. Po wstawieniu tam wybranej liczby (maksymalnie $6F) można delektować się swobodą finansową. Uprzedzam jednak, że taki zabieg łatwo uzależnia. Kiedy raz zwiększysz sobie konto, trudno jest nie zrobić tego ponowie. No cóż, cywilizowani ludzie tak chyba nie postępują, lecz z drugiej strony miły jest widok pierwszej eksplozji nuklearnej w roku 1745 - czyli 200 lat wcześniej.

HANAKEN D.G.

DODATEK:
WYKAZ I OPIS CUDÓW ŚWIATA.

APOLLO PROGRAM - umożliwia konstruowanie statków kosmicznych oraz czyni widocznymi wszystkie miasta na mapie:

COLOSSUS - polepsza handel w posiadającym go mieście (działa do odkrycia ELECTRICITY)

COPERNICUS

OBSERVATORY - przyspiesza produkcję w posiadającym je mieście (do odkrycia AUTOMOBILU)

CURE FOR CANCER - powiększa ilość szczęśliwych ludzi w populacji

DARVIN'S VOYAGE - powoduje natychmiastowe odkrycie dwóch wynalazków

GREAT LIBRARY - pozwala zdobyć dwa wynalazki, które posiadły inne cywilizacje (do odkrycia UNIVERSITY)

GREAT WALL - powoduje, że inne cywilizacje zawsze pierwsze proponują pokój (do odkrycia prochu - GUNPOWDER)

HANGING GARDENS - powiększa liczbę szczęśliwych ludzi w każdym mieście (do odkrycia INVENTION)

HOOVER DAM - dostarcza hydroenergii do wszystkich miast na kontynencie (redukuje zanieczszczenie)

ISSAC'S NEWTON

COLLEGE - powiększa efekt z posiadania LIBRARY i UNIVERSITY (do odkrycia NUCLEAR FISSION) J.S.

BACH CATHEDRA - zmniejsza liczbę nieszczęśliwych ludzi do 20<188> na miasto

LIGHTHOUSE - powiększa o 1 liczbę ruchów po morzu (do odkrycia MAGNETISM)

MAGELLAN'S

EXPEDITION - powiększa o 1 liczbę ruchów po morzu

MANHATTAN PROJECT - umożliwia skonstruowanie broni nuklearnej

MICHELANGELO'S

CHAPEL - powiększa efekt CATHEDRAL (do odkrycia COMMUNISM)

ORACLE - powiększa efekt TEMPLE (do odkrycia RELIGION)

PIRAMIDS - umożliwia zmianę rządu bez dwóch tur anarchii oraz czyni osiągalnymi wszystkie formy rządów (oba efekty do odkrycia COMMUNISM)

SETI PROGRAM - przyspiesza produkcję we wszystkich miastach o 50%

SHAKESPEARE'S

THEATRE - czyni zadowolonymi wszystkich ludzi w mieście (do odkrycia ELECTRONICS) UNITED NATIONS - inne cywilizacje zawsze pierwsze oferują pokój

WOMEN'S SUFFRAGE - zmniejsza liczbę nieszczęśliwych ludzi w mieście - proporcjonalnie do ilości oddziałów pozostających poza miastem (ale tylko,gdy ustrój polityczny to republika lub demokracja)

CRAZY BOYS & SKYLIGHTS C64 COPY PARTY

Od kilku miesięcy było już wiadomo, że na tym party dopisze frekwencja. Zewsząd docierały pogłoski o przygotowywanych specjalnie na ten zlot programach demonstracyjnych. Ale przejdźmy do szczegółów. Copy party zostało zorganizowane dzięki operatywności członków dwóch grup - *Skylight* i *Crazy Boys*. Odbyło się ono w dniach 28 i 29 grudnia 1992 w Szczecinie. Na potrzeby uczestników zaadoptowano salę gimnastyczną, korytarze i kilka innych pomieszczeń Liceum Ogólnokształcącego nr 5. Czas trwania: od 10 rano 28 grudnia (poniedziałek po świętach Bożego



Narodzenia), przez całą noc aż do późnych godzin nocnych dnia następnego.

Wielu obecnym grupom nie udało się skończyć swoich dem (uczciwie mówiąc, mało kto próbował to robić na party...). Wiele osób zobaczyło się pierwszy raz na oczy, znając się tylko uprzednio z oglądania nawzajem swoich programów lub kontaktów telefoniczno-listowych. A było trochę znanych grup (oczywiście, poza organizatorami) jak np. *Elysium*, *Taboo* czy też *Inflexion*. Pojawiły się także "nowe twarze", których przykładem może być grupa *Diadem*. Pomimo, że party było z założenia tylko dla posiadaczy Commodore 64, nie przeskodziło to



jak widać użytkownikom Amigi w pojawieniu się na tej imprezie.

Główną salą była oczywiście sala gimnastyczna. Efekt był wstrząsający zwłaszcza pod koniec drugiego dnia - góry śmieci i błota na parkiecie świadczyły o wzmożonym ruchu pieszych pomiędzy swoimi stanowiskami komputerowymi. Jak powiedział mi mój redakcyjny (reakcyjny) kolega - Krzysiek Kobus (absolwent tegoż właśnie liceum) - "O Boże, a nas to nawet w butach na salę nie wpuszczano...". Oprócz tego był jeszcze Sleeping Room (czyli pokój do testowania śpiworów) i tzw. bufet, w którym można było dostać coś do picia i zjedzenia (nawet i niczego sobie kiełbaskę) po naprawdę umiarkowanych cenach.

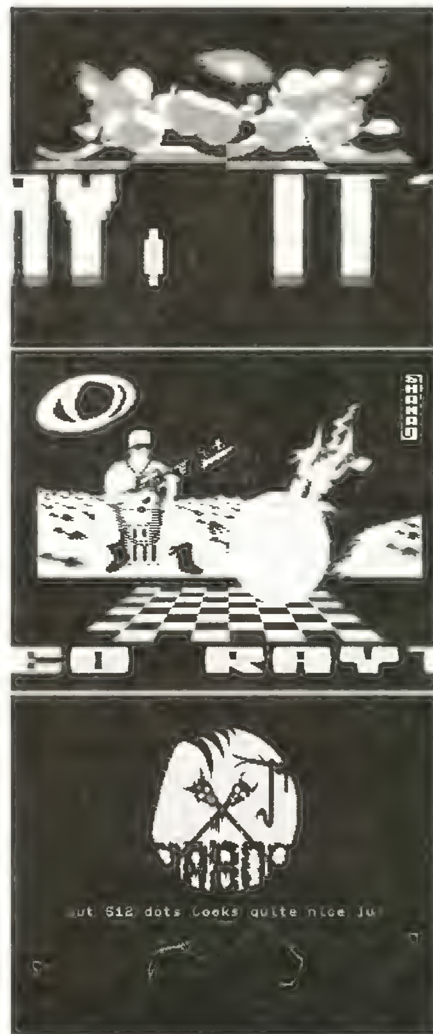
W dobie konkurencji na rynku, na party pojawiły się dwie ekipy telewizyjne, jedna państwowa a druga prywatna. W praktyce obie rzuciły się na imprezę jak "szczerbaty na suchary", kręcąc po prostu ustawionych w rządek organizatorów i krótkie migawki z party.

Na potrzeby democompetition organizatorzy uwinęli się, że aż miło i zorganizowali tzw. Big Screen, czyli duży projektor do obrazu telewizyjnego, na którym oglądane były startujące w konkursie programy demonstracyjne. Bezapelacyjnie pierwsze miejsce zajęło zapowiadane wcześniej demo **The Place in Space** grupy **Taboo**. Ponieważ bardziej znane grupy



praktycznie nie zmieściły się w ramach czasowych imprezy z oddaniem swoich produkcji, poważnych konkurentów praktycznie nie było.

Samo demo (bo warto o nim powiedzieć kilka słów) zostało zrobione bardzo starannie i na wysokim poziomie technicznym. Zwłaszcza bardzo wysoko należy ocenić grafikę wektorową (wypełnioną i bardzo szybką;



m.in. tzw. TV-BOX), liczony "na poczekaniu" Ray Tracing, Sin-Plot-Scrolle itd. Moim osobistym zdaniem grupa Taboo wybiła się na czołowe pozycje w Polsce i Europie za sprawą swoich dwóch uzdolnionych programistów: **MMS** i **KM**.

Pozostaje mieć tylko nadzieję, że tradycja Copy Party w Polsce będzie kontynuowana i dostarczy wielu ambitnym osobom bodźców do twórczej pracy.

Paweł "Polonus" Sołtysiński

CZEŚĆ! Grupa "FUCKED DOG" **nawiąże** kontakt z innymi grupami amigowskimi.
Nasz adres:
ul. Woźniaka 26/8
41-902 Bytom tel 815-119

Grupa ATLANTIC (C-64) **poszukuje** członków wszelkich specjalności, a szczególnie koderów. Napiszcie!
Tomasz Augustyn
ul. Makarskiego 11/11
49-300 Brzeg

Sprzedaż oprogramowania do Amigi, duży wybór, niskie ceny. Katalog gratis. Sprzedam komputer Amoga 600 (gwarancja).Cena do uzgodnienia.
Przemysław Mikosz
ul. Buczka 27/12
43-300 Bielsko-Biala
tel 495-37

Sprzedam A500, 2.6Mb, RAM, (gwarancja), dodatkowe akcesoria. Cena 6.6 mln. tel. 527-318
Tomasz Rachwal
ul. Romera 61/10 Szczecin

Sprzedam nową stację do C-64, FLOPPY 9900 (GWARANCJA), 20 dysków i FINAL II za 2 mln.
Daniel Sier
ul. Kaczeńców 8
44-240 Żory-Rój

C-64 - gry i użytki (kaseta/dysk). Informacje: koperta + znaczek.
Mariusz Listowski
ul. Sobieskiego 17/8
76-200 Slupsk

Bezpłatna **wymiana** programów na Amigę.
Krzysztof Nadowski tel 221-90
ul. Niepodleglosci 4/4
38-300 Gorlice

Sprzedam opis uzyskania niesmiertelnosci na C-64. Cena 20 tys. + koszt wysyłki
Kamil Wiatrowski
ul. Jana z Kolna 9/25
72-600 Świnoujście

Umiesz coś na C-64. Napisz. Założę grupę.
Patryk Dziurla
lu. Konopnickiej 3/21
64-200 Wolsztyn

Sprzedam: C-64, drukarkę, magnetofon, Final III, Black Box - 3mln.
Piotr Witomski
ul. Hetmańska 56/71
60-219 Poznań

Wymienię programy użytkowe C-64, Amiga. Pilnie **poszukuję** programow pakujących na C-64. Wszystkie dla stacji dysków.
Jędrzej Chmielewski
ul. Kąkolowa 7/33
85-811 Bydgoszcz

Sprzedam A-500 (gwarancja), monitor 1084S (gwarancja), dyski, dodatki, literatura za 10,5 mln.
Grzegorz Makowski tel 525-63
ul. Parkowa 1/34
78-300 Świdwin

Sprzedam gry i programy na C-64. Cena dysku 15 tys., kaseta 20 tys. Spis gratis!
Wojciech Tomaszewski
ul. Strażacka 7
68-205 Żary

Grafik (Amiga 500) **skontaktuje** się z początkująca grupą.
Tomasz Pacyna
ul. Poznańska 3
64-200 Wolsztyn

Sprzedam ponad 1000 programów (kaseta) na C-64. Informacje: koperta + znaczek.
Tomasz Dąbkowski
Racław 15
66-432 Baczyna

Sprzedam Action Replay MK II - gwarancja do 12.1993, instrukcja w języku polskim - 1.5 mln. Pilne!
Piotr Laszczyk tel 70-444
Koscielisko 976
34-511 Kościelisko

DEATH KISS (Amiga) **poszukuje** muzyków, grafików, koderów do tworzenia gier oraz dem. Pilne! Kontakt listowny.
DEATH KISS
ul. W. Skotnica 20/40
41-400 Mysłowice

Adresy firm oraz grup komputerowych w Polsce i na całym świecie. Cena 10 tys.
Patryk Dziurla
ul. Konopnickiej 3/21
64-200 Wolsztyn

Coder (C-64) poszukuje grupy do której mógłby wstąpić.
Marcin Kuciapski
ul. Traugutta 26/22
99-320 Żychlin

Sprzadam tanio oprogramowanie na C-64 (kaseta i dysk). Katalog po otrzymaniu koperty i znaczka.
Marcin Jastrzębski
ul. Reymonta 43 A/15
96-100 Skierniewice

Poszukuję C-64 bez obudowy i klawiatury - sprawny moduł.
Andrzej Pappelbaum
ul. Modrzewskiego 2/18
86-300 Grudziądz

HELP! **Szukam** "DOWNLOAD" do MPS - 1230 z C-64.
Leszek Cepil
Os. Sikorskiego 14/45
28-100 Busko-Zdrój

Sprzedam C-128 + monitor + wzmacniacz współpracujący, ok. 500 programow, literatura, czasopisma, 2 joysticki, Final 3 BBV7. Cena ok. 3 mln.
Tomasz Wojtyniak tel 776-279
Os. Czecha 45/3
61-288 Poznań

Programy - Amiga -(gry, użytki, dema). 6 tys./szt!!! Koperta + znaczek.
Daniel Pastuszko
ul. Iberyjska 4/2
02-764 Warszawa

Wymienię lub **kuplę** użytki (nowości) na C-64 oraz Amigę 0.5 Mb). Poszukuję cruncherów sekwencyjnych (C-64).
Jedrzej Chmielewski
ul. Kąkolowa 7/33
85-811 Bydgoszcz

Grupa "DUET" - **szukamy kontaktów.** Wymiana gier, dem i użytkow a także doswiadczeń na C-64. Nawiążemy kontakty ze znającymi i uczącymi się asemblera.
Łukasz Nosal
ul. Złota 11 A/8
78-100 Kołobrzeg

Wzmacniacz gitarowy Regent 50W, mikrofon MDU 232+ kabel, 2 kpl strun ROCK I **zamienlę** na stację 1541 II.
Krzysztof Najborowski
Os. Batorego 4/56
60-687 Poznań

Uwaga grupy (C-64)! Muzyk (VERY GOOD IN HIS JOB) **nawiąże** kontakt bądź wstąpi do grupy.
Marcin Burkot
ul. Redutowa 2/80
22-400 Zamość

Sprzedam magnetofon do C-64, cartridge Black Box V.8, 16 kaset (gry, użytki)
Remigiusz Felcenloben
tel. 771-453
ul. Struga 15/26
73-110 Stargard

Wymienię oprogramowanie (ok. 600 pozycji) na C-64. Kaseta, gry i użytki.
Krzysztof Obrycki
ul. Zielona 16
18-421 Piątnica

Amigę 500(1Mb), (gwarancja) + 100 dysków - **sprzedam**. Cena 5.6 mln.
Tomasz Czyżewski tel 168-344
ul. Prusa 9/19
18-400 Łomża

Listingi

AMOS

```
Global A$
SETUP
Procedure SETUP
    Screen Open 0,640,256,4,Hires
    Paper 0
    Cls
    Hide                              jak zwykle nale y zrobi- "nastrĚj"
    Curs Off
    Paper 1
    Print At(0,25);Space$(80)
    Paper 0
End Proc
STAR:
Restore DANE
For F-1 To 3 : Rem zamiast trĚjki mozna wstawi- ilod- linii Data z tekstem
    Read A$
    SKROL
Next
Goto STAR
Procedure SKROL                          w+ađciwa procedura sprowadzaj-ca si-
    For F=640 To(-8*Len(A$)) Step -2     do drukowania napisu z przesuni-ciem
        Wait Vbl                         o 2 piksele.
        Text F,206,A$
    Next F
End Proc
DANE:
Data " To jest przyklad prostego scroll'a w Amosie. "
Data " Pamietaj, ze tekst nie moze byc zbyt dlugi. "
Data " Amos jednak nie jest do pisania dem !!! "
```

Mapa pamięci

```
Screen Open 0,640,256,8,Hires
Paper 0
Pen 2                           otwieramy ekran i tworzymy "mi+-
Cls                            atmosfer-" tzn. ciemne t+o, bez migaj-
Hide                            cego kursora, pointera itp.
Curs Off
Global WYNIK
OZDOBY
STAR:
Restore DANE
Read ILE           tu dopiero wida- jak wa na jest dana o ilo~ci pyta¤
For F=1 To ILE
Cls
   Read P$,A$,B$,C$,D$,WPLUS$,WMINUS$  -odczyt danych
   Pen 5
   Locate 0,10 : Centre P$
   Pen 2
   Locate 0,14 : Centre A$   drukujemy pytania
   Locate 0,16 : Centre B$    i odpowiedzi
   Locate 0,18 : Centre C$
   Locate 0,20 : Centre D$
   Do
      I$=Inkey$
      If I$>"0"                                          g+Ëwna p-tla
         If I$<"5"
            If I$=WPLUS$
               WYNIK=WYNIK+1         sprawdzamy jak-
            End If                  wybrano odpowied+
            If I$=WMINUS$             i modyfikujemy wynik
               WYNIK=WYNIK-2
            End If
            Exit
         End If
      End If
   Loop
Next
KONIEC
Procedure KONIEC
   Cls
   If WYNIK<=0
      Locate 0,10 : Centre "Lepiej nie m˘wi- o Twoim wyniku..."
      Print
      Print
      Centre "Pewnie stara+e~ si-, ale niewiele z tego wysz+o. SprËbuj jeszcze raz !"
      Locate 0,20
      Pen 3
      Centre "Wci~nij dowolny klawisz..."
      Pen 2
      Wait Key
   Else
      Locate 0,10 : Centre "Brawo ! "
      Print
      Print
      Print "                    Odpowiadaj-c na pytania uzyska+e~";WYNIK;" pkt."
      Locate 0,20
      Pen 3
      Centre "Wci~nij dowolny klawisz..."
      Pen 2
      Wait Key
   End If
   Cls
   Locate 0,14
   Centre "Grasz jeszcze raz ? ( T/N )."
   Do
      I$=Inkey$                  pytamy si- o powtËrzenie testu
      If I$="n"                 i czekamy na odpowied+
         End
      End If
      If I$="t"
         Exit
      End If
   Loop
End Proc
Procedure OZDOBY
   Set Rainbow 0,0,200,"","","(1,1,15)(1,-1,15)"
   Set Rainbow 1,0,200,"(1,1,15)(1,-1,15)","",""   stawiamy bary ku
   Rainbow 0,0,40,32                                uciesze gawiedzi.
   Rainbow 1,0,256,32
End Proc
Goto STAR        - restartujemy program od etykiety STAR.

DANE:
Data 2           - tu wstawiamy ilo~- pyta¤.
Data "pytanie1 ?","odp11","odp12","odp13","odp14","2","4" \  tu dopisujemy
Data "pytanie2 ?","odp21","odp22","odp23","odp24","3","4" /     dane
```

"CHARBLASTER II "

$0801-$10E6

```
:0801 0B 08 90 06 9E 32 30 34 (1E)
:0809 39 00 A0 00 78 E6 01 B9 (B5)
:0811 51 10 99 FA 00 C8 D0 F7 (55)
:0819 4C 00 01 09 60 E9 30 60 (3A)
:0821 AF BD F3 8E 08 08 08 E7 (2B)
:0829 92 51 42 4C 41 E5 C7 B7 (27)
:0831 20 B9 DA D7 5D B1 C6 7A (E6)
:0839 31 FE 5D 65 A0 2F E1 5E (6A)
:0841 52 A0 52 79 17 52 D7 79 (BD)
:0849 28 57 29 50 4F AE 50 4E (21)
:0851 55 1E 71 BC E1 43 29 31 (CB)
:0859 39 39 32 8F 35 FF 59 BB (28)
:0861 2F 4D 4D 4F 79 E6 81 20 (9D)
:0869 F8 2D BE EB 42 3F CA 67 (2B)
:0871 E3 FB 6B 38 15 E1 C1 18 (29)
:0879 BE 8A 20 BF E3 A9 0B D7 (19)
:0881 02 8D 21 D0 4E C2 48 F6 (02)
:0889 47 59 7A 61 A4 97 95 40 (4D)
:0891 03 C3 72 F1 A9 8C A5 64 (74)
:0899 68 8D F3 A9 A2 8B C5 43 (87)
:08A1 B6 17 78 3C 58 8D 62 10 (19)
:08A9 C9 58 89 1D A9 45 2C A9 (A0)
:08B1 4C 76 25 10 86 EF A9 B8 (37)
:08B9 5D F4 94 5F E3 90 34 87 (B1)
:08C1 E0 10 01 A8 E1 10 90 F1 (A9)
:08C9 8E 0F 01 E0 00 F0 E8 A9 (40)
:08D1 CA 5E 48 11 E3 4E D0 84 (8E)
:08D9 89 B9 1E E9 A0 0C 6A 4E (98)
:08E1 2B A9 40 39 DC 10 F0 8D (AE)
:08E9 64 10 8C 65 10 AD 50 03 (53)
:08F1 29 0F 09 30 85 37 20 04 (FE)
:08F9 A6 AF 27 20 B3 14 A2 0F (D7)
:0901 A9 CA A4 D2 FF 44 3A 20 (A4)
:0909 0B 08 A9 32 A0 C1 F6 BB (28)
:0911 20 66 10 A9 E3 C8 7A A8 (8F)
:0919 F0 DA F8 20 47 11 3F 05 (E5)
:0921 F0 FB C9 11 D0 0C A6 FC (71)
:0929 15 6D B5 FB B0 F0 E6 29 (CE)
:0931 A3 91 D0 90 C2 FC F0 E3 (09)
:0939 C6 FC 4C B3 59 02 5F F0 (92)
:0941 A6 27 96 D0 7C AD A7 25 (73)
:0949 2C 6F 65 4A 96 C9 A9 23 (0E)
:0951 8D FB 05 1A EC 87 C9 8D (01)
:0959 FD 05 9B CA A2 08 C0 D7 (B4)
:0961 0F 08 15 1A A3 A2 81 A0 (B2)
:0969 4E F0 8F 54 8F 85 9B CF (3B)
:0971 E8 58 C2 9E 7A 6B A9 24 (73)
:0979 8D 09 A5 C2 2E 2A 02 68 (48)
:0981 49 F3 A5 C1 20 BE 11 8D (9F)
:0989 04 06 8C 03 E6 B6 90 11 (8C)
:0991 1B 2F B9 E8 05 9C 91 C0 (96)
:0999 13 B0 F6 1B 5A 20 A3 0B (B2)
:09A1 20 0D A0 DB 58 B6 09 A5 (93)
:09A9 FD F0 F9 A9 06 8D 86 02 (44)
:09B1 72 F7 D4 E5 78 A8 89 9C (11)
:09B9 0D C1 03 E2 0E C5 8B 04 (B3)
:09C1 BD 9C 0F 9D 36 05 D9 51 (03)
:09C9 EB FF 06 06 CA 06 39 E8 (CA)
:09D1 86 FE E6 FD 7D 95 D6 65 (F3)
:09D9 5C A0 15 83 44 C5 FE F1 (35)
:09E1 75 8F 4C 9D 43 0C 8A ED (9A)
:09E9 9C 31 F0 1B 08 1B 93 A2 (0B)
:09F1 C8 4C BC 0A 29 06 55 A0 (FA)
:09F9 19 B1 20 E9 8D 1F C8 E8 (B4)
:0A01 E0 04 57 30 E3 A0 1F 20 (C0)
:0A09 16 08 A6 FE CA 9D 76 1B (D5)
:0A11 98 9D 14 85 22 FF D2 C5 (C7)
:0A19 FF 4B 9E A9 71 F7 AC 0F (61)
:0A21 C6 FD 14 49 3C EA AC FD (8F)
:0A29 8F 91 3A FD 8F 90 0A C4 (17)
:0A31 DB 4C AC DF 66 AD E8 E4 (B2)
:0A39 CA 5C E4 7F 90 C9 07 90 (A4)
:0A41 CF 0A 02 B0 06 27 8D C9 (1F)
:0A49 E0 90 4F 14 36 03 A9 43 (67)
:0A51 A0 0B 63 A9 35 39 D6 05 (3F)
:0A59 93 1A 84 74 52 21 64 74 (42)
:0A61 C2 45 27 53 24 41 FA B2 (18)
:0A69 20 D6 46 56 CF 53 02 2C (D4)
:0A71 2C 5B 59 8D 06 4D 5B D4 (A5)
:0A79 F8 52 7D C6 1E 16 51 62 (0F)
:0A81 2C 0D BC 9D 8A 44 5B C3 (58)
:0A89 43 93 8B 54 20 49 54 29 (D7)
:0A91 47 35 00 85 02 E9 54 49 (74)
:0A99 80 D8 01 97 0D F0 1E C9 (2D)
:0AA1 30 90 F5 C9 47 B0 F1 C9 (60)
:0AA9 41 B0 04 C9 3A B0 E9 26 (55)
:0AB1 FB 6A 9A 04 E8 8A 29 03 (63)
:0AB9 4C A5 0B A6 02 BD 81 4B (69)
:0AC1 F4 80 01 06 41 A0 0A DB (FD)
:0AC9 E8 70 58 14 3E 5B 43 57 (D8)
:0AD1 D9 4C 78 B4 53 48 38 6C (BB)
:0AD9 F2 53 0E 27 45 6F 08 B5 (14)
:0AE1 4F 55 54 D3 C8 6E 94 24 (D4)
:0AE9 35 30 21 A8 8E 6E 3E 44 (B7)
:0AF1 CB 55 50 98 27 31 36 21 (2B)
:0AF9 7E 2D A6 D3 2C 19 AA 29 (79)
:0B01 93 45 56 52 CC 41 F8 27 (F5)
:0B09 30 37 45 30 A8 97 38 2A (EB)
:0B11 30 16 2C 73 F4 23 52 57 (54)
:0B19 54 63 72 53 D7 B1 4A 4A (8F)
:0B21 55 4D 50 51 0D 00 46 0C (DA)
:0B29 40 45 24 30 31 20 56 41 (41)
:0B31 4C 55 45 3A 20 24 33 37 (7E)
:0B39 8B 6F 45 12 53 92 45 49 (FA)
:0B41 2F 12 43 92 4C E3 EA 3A (B4)
:0B49 43 7D 84 22 DC 8A EB 22 (AA)
:0B51 41 4D 5F 34 3A 76 8F 45 (1B)
:0B59 76 87 37 54 48 2F EE 20 (E1)
:0B61 4F 4B 3F 20 28 59 2F 4E (25)
:0B69 29 3A 59 C3 DB 63 38 39 (11)
:0B71 12 DB 5C 4D 52 7F CA 28 (E6)
:0B79 20 26 13 B6 64 24 85 94 (10)
:0B81 30 42 41 52 2E 8C 02 95 (2F)
:0B89 61 99 67 E1 53 0E 5A 42 (59)
:0B91 44 3A 0D 68 EF 94 5F 6E (47)
:0B99 4E 45 57 21 03 32 90 9A (00)
:0BA1 4F 52 59 3B 4A 91 DA D8 (24)
:0BA9 02 14 C5 4B F2 47 D5 3D (78)
:0BB1 98 8C 38 45 53 45 D1 43 (34)
:0BB9 54 88 BE B8 B4 62 52 45 (78)
:0BC1 54 55 52 AC 16 41 47 41 (5D)
:0BC9 49 4E 20 2D B3 AC 58 4C (1C)
:0BD1 4F 43 20 41 44 44 59 61 (78)
:0BD9 0D A9 38 85 9A 42 8A 9D (3B)
:0BE1 00 4A B6 FA EE 40 23 E3 (BD)
:0BE9 F5 DD 4A 0C 05 A0 FF 28 (C3)
:0BF1 BE 13 54 70 2A 8D 4E 88 (1E)
:0BF9 B1 AE E6 DC C8 91 AE E6 (73)
:0C01 97 32 AE FA B1 AF C6 05 (1B)
:0C09 47 5D A2 D0 07 A5 AC 48 (31)
:0C11 A5 AD 48 C8 98 B9 93 99 (2F)
:0C19 53 88 FE 60 C5 3A 56 86 (C1)
:0C21 AA E7 00 D7 F0 03 FE D9 (7D)
:0C29 A9 41 90 E7 0C 85 AD 68 (01)
:0C31 85 AC 8C F5 08 4B CF 06 (55)
:0C39 50 8B 08 38 5B EE 9F D0 (D7)
:0C41 F0 8E 98 53 8E D1 B1 C9 (38)
:0C49 D2 6F 2D 07 85 FC 09 A1 (70)
:0C51 AC 8D 85 9E D1 AC D0 D9 (BF)
:0C59 87 F9 88 98 49 FF A8 C4 (F5)
:0C61 C9 72 01 C8 8C 7A 0E 45 (5F)
:0C69 CD E1 A1 36 C0 FD 90 09 (A5)
:0C71 4C 17 D0 A0 AD C0 03 16 (8D)
:0C79 CA 02 45 0E 20 15 04 AC (F4)
:0C81 C1 07 82 9D F1 04 5B FA (70)
:0C89 4C C7 03 39 DD 52 DC FB (75)
:0C91 F9 4D FC 60 82 DF 20 DB (20)
:0C99 FC 20 D1 FC B0 05 6A 2D (20)
:0CA1 60 00 71 C3 FB 8D D1 29 (A0)
:0CA9 3B 4F 2B 61 E3 B9 90 ED (B0)
:0CB1 91 FB C8 9D B8 98 18 65 (08)
:0CB9 FB 4E A5 1C CE FC A0 4D (71)
:0CC1 B9 B2 05 99 01 08 B2 AD (D8)
:0CC9 5D AD 2D B0 22 08 AD 2E (D8)
:0CD1 04 8D 23 98 C5 AE 38 E9 (81)
:0CD9 07 8D 2E 08 B0 D4 7A AF (E6)
:0CE1 A5 AF 8D 2F BB 82 37 8D (EF)
:0CE9 46 08 9A 8E CB C1 BD FE (E9)
:0CF1 04 F0 0D 95 C0 23 07 AE (8F)
:0CF9 94 E8 D0 EE AD 51 03 8D (55)
:0D01 3B 06 AD 7F 8D 00 DC AD (85)
:0D09 01 DC C9 EF E7 76 D1 CA (34)
:0D11 A9 0B 9D 98 DB CA 10 FA (57)
:0D19 B7 0E 92 60 83 AD 0F 30 (B5)
:0D21 9E 10 FB 05 CF C1 C9 6D (69)
:0D29 2F 05 02 15 36 0C 16 66 (E9)
:0D31 09 C3 B1 1A 43 EE 2E 20 (6D)
:0D39 D2 FF CE D2 03 7B 10 1B (01)
:0D41 FE BE 87 FB B4 87 FC 90 (5B)
:0D49 DD E4 8A 7E 04 53 50 41 (CF)
:0D51 43 84 AD 54 F2 81 53 41 (0D)
:0D59 56 45 9C 7A DA 94 E0 9F (D4)
:0D61 A9 08 60 4C 86 F0 F8 38 (3D)
:0D69 C9 A2 1B E6 00 0A 2E 86 (1A)
:0D71 3A BD 00 06 F0 09 C8 C0 (A8)
:0D79 10 B0 08 E8 4C 27 05 C0 (37)
:0D81 00 D0 13 FB 8E 84 39 1B (98)
:0D89 DD EC 4C 92 26 A5 39 A6 (D2)
:0D91 3A A0 06 D8 2B 01 26 A6 (A1)
:0D99 38 BD 20 39 4A FD BD 30 (A7)
:0DA1 01 85 FE C4 3A 8D C9 C0 (B2)
:0DA9 78 E6 46 1B 91 FD C6 01 (6D)
:0DB1 E6 FD D0 04 E6 FE F0 25 (48)
:0DB9 F1 12 F0 E5 19 81 A5 AE (B5)
:0DC1 05 AF F0 0B 38 C8 46 E5 (07)
:0DC9 9E E5 AF 90 08 A5 FD 85 (A4)
:0DD1 AE A5 FE 85 AF E6 38 4C (9B)
:0DD9 18 05 20 AA 05 4C 97 05 (3A)
:0DE1 40 BF 65 4C D6 66 0B 08 (2A)
:0DE9 CF 07 9E 32 30 35 39 00 (32)
:0DF1 E6 42 78 BE 82 07 96 81 (9E)
:0DF9 88 30 F8 4C 24 00 A5 21 (45)
:0E01 1B 5F 0A 22 C6 21 AD 5A (BD)
:0E09 13 60 B1 21 91 02 5D 10 (6D)
:0E11 F9 CA 9D A8 1B D0 02 C6 (C8)
:0E19 2E 6F 9B 02 30 C9 1E D0 (04)
:0E21 EB 23 1D A8 F0 82 00 C0 (13)
:0E29 37 D0 E3 84 01 58 4C 74 (90)
:0E31 A4 BB 18 80 59 AA 13 05 (07)
:0E39 AF 61 FB 5F 26 F6 A9 21 (4E)
:0E41 25 94 29 FB C8 90 A2 60 (B9)
:0E49 A0 A3 20 BD FF D9 76 55 (84)
:0E51 A2 08 20 BA A5 C0 FF A2 (1B)
:0E59 C7 0C C6 77 01 38 06 A5 (1B)
:0E61 90 D0 49 A8 00 A3 42 F0 (3A)
:0E69 3C A1 7C A0 00 10 1A A0 (FF)
:0E71 C8 E0 35 51 D0 F5 66 FC (62)
:0E79 60 F0 F9 C9 50 D0 15 E6 (09)
:0E81 FB B9 D8 2D 96 91 20 71 (F4)
:0E89 C2 F8 BF 0B 20 CF FF D0 (A5)
:0E91 FB 4C A3 10 A9 01 20 C3 (A6)
:0E99 FF 20 CC FF A5 FB F0 01 (F9)
:0EA1 60 A9 1A A0 11 20 1E AB (6E)
:0EA9 A9 00 85 C6 20 E4 FF C9 (40)
:0EB1 20 D0 F9 68 68 4C 82 09 (B0)
:0EB9 13 87 F0 05 6C 0D 03 20 (4B)
:0EC1 4E 4F 20 50 52 47 20 46 (AF)
:0EC9 49 4C 45 53 21 20 43 48 (4D)
:0ED1 41 4E 47 45 20 44 49 53 (74)
:0ED9 4B A8 BF 2E 00 40 74 46 (53)
:0EE1 A9 4B 21 A6 FC F0 0E D6 (C7)
:0EE9 64 E4 A9 F8 A2 0F A8 A9 (62)
:0EF1 04 85 23 A0 1F 51 EC B0 (6B)
:0EF9 29 3F 91 22 88 10 F7 59 (7A)
:0F01 A3 70 A5 22 68 28 85 22 (B5)
:0F09 99 1B 23 0E DC 60 A5 FC (77)
:0F11 4A 29 06 AA A9 40 69 4A (72)
:0F19 85 21 A5 20 18 69 20 85 (54)
:0F21 20 90 02 E6 21 CA D0 F2 (AF)
:0F29 A0 01 B1 20 C9 22 F0 06 (E6)
:0F31 99 80 03 C8 D0 F4 88 8C (E2)
:0F39 90 03 60 48 24 C0 17 4A (43)
:0F41 C0 06 A8 68 29 0F 20 CE (2B)
:0F49 11 60 C9 0A B0 03 09 30 (ED)
:0F51 60 E9 09 60 EA C8 C8 C8 (27)
:0F59 A9 00 85 10 A6 FB 30 0D (B8)
:0F61 06 FA 2A 26 10 C6 FB 88 (91)
```

```
:0F69 D0 F2 AA 18 60 48 A1 FF (79)
:0F71 85 FA A2 07 86 FB E4 FF (AF)
:0F79 A6 FE D0 02 C6 FF C6 FE (D4)
:0F81 68 90 DD E0 E7 D0 D9 A9 (C9)
:0F89 37 85 01 58 4C 01 08 A9 (3E)
:0F91 10 E6 11 24 11 85 04 AA (1E)
:0F99 BC D8 07 20 24 07 A6 04 (31)
:0FA1 7D BA 07 48 A5 10 7D C7 (12)
:0FA9 07 A8 68 A6 02 D0 08 C0 (01)
:0FB1 00 D0 04 C9 01 F0 D8 18 (DD)
:0FB9 65 FC AA 98 65 FD 85 49 (55)
:0FC1 A4 11 F0 20 8A 38 E5 11 (B3)
:0FC9 B0 03 C6 49 38 85 48 A5 (5A)
:0FD1 FC E5 11 B0 02 C6 FD 85 (5A)
:0FD9 FC B1 48 88 91 FC C6 01 (6D)
:0FE1 EE 20 D0 E6 01 98 D0 F1 (F3)
:0FE9 60 28 07 0C 0E 29 00 08 (69)
:0FF1 1A 3C 7F FF 7F 7F 00 10 (00)
:0FF9 20 40 80 00 00 00 00 00 (28)
:1001 00 01 02 00 00 00 00 00 (19)
:1009 01 03 07 0B 03 04 05 06 (DB)
:1011 07 07 08 09 04 04 05 06 (F1)
:1019 07 09 0A 0A 01 57 29 4C (16)
:1021 A0 E1 B2 B2 28 A9 04 11 (D3)
:1029 BF 41 60 AD FF 84 21 1D (30)
:1031 65 00 97 1B 19 00 0A 2F (12)
:1039 00 E4 C9 BD 08 15 C1 95 (F5)
:1041 C1 D5 73 02 90 E1 C9 38 (72)
:1049 A4 C1 60 4F 7B B0 03 69 (BF)
:1051 BE 03 D8 11 55 0F B9 51 (8B)
:1059 0F 99 1C 07 C8 D0 F7 20 (A3)
:1061 23 07 F0 46 20 23 07 D0 (AD)
:1069 30 20 22 07 69 02 C9 04 (23)
:1071 90 27 D0 07 20 23 07 69 (D6)
:1079 04 D0 1E 20 21 07 69 06 (E5)
:1081 C9 0D D0 11 C8 20 21 07 (EF)
:1089 69 0D C9 0D D0 07 A0 04 (65)
:1091 20 24 07 69 25 EE 00 04 (2F)
:1099 EA 85 11 A6 FE A5 FF 20 (35)
:10A1 8A 07 A5 49 85 FF A5 48 (B2)
:10A9 85 FE 20 23 07 85 02 F0 (F5)
:10B1 14 0A 2C A9 03 85 11 20 (B5)
:10B9 21 07 A6 02 D0 02 69 08 (2D)
:10C1 20 61 07 F0 9A 20 23 07 (77)
:10C9 F0 E9 20 22 07 69 04 C9 (80)
:10D1 06 90 E2 D0 07 20 22 07 (F6)
:10D9 69 06 D0 D9 A0 05 20 24 (70)
:10E1 07 69 0A D0 D0 20 63 6F (25)
```

"RAMDISK "

$0801-$0DA3

```
:0801 0B 08 C9 07 9E 32 30 36 (DD)
:0809 31 00 00 00 A9 80 85 FB (0A)
:0811 A9 9A 85 FC A9 32 85 FD (79)
:0819 A9 08 85 FE A2 06 A0 00 (0F)
:0821 B1 FD 91 FB C8 D0 F9 E6 (3A)
:0829 FE E6 FC CA D0 F2 4C 80 (E7)
:0831 9A 20 89 9A 20 66 9B 4C (B7)
:0839 C8 9A A9 00 8D F0 9F A9 (3A)
:0841 13 8D F1 9F A9 A1 8D 02 (C3)
:0849 A0 60 48 8A 48 98 48 A9 (E9)
:0851 37 85 01 A9 7F 8D 0D DD (4D)
:0859 AC 0D DD 10 03 4C 72 FE (E3)
:0861 20 BC F6 20 E1 FF D0 F5 (1A)
:0869 20 C8 9A 20 A3 FD 20 18 (2C)
:0871 E5 A9 01 8D 86 02 6C 02 (95)
:0879 A0 A0 1F B9 30 FD 99 14 (4F)
:0881 03 88 10 F7 A9 99 A0 9A (BB)
:0889 8D 18 03 8C 19 03 A9 20 (B5)
:0891 85 73 A9 2B 85 74 A9 9B (73)
:0899 85 75 A9 EA 85 76 85 77 (6B)
:08A1 85 78 A9 54 A0 9E 8D 30 (98)
:08A9 03 8C 31 03 A9 0E A0 9D (54)
:08B1 8D 32 03 8C 33 03 A9 80 (93)
:08B9 A0 9A 85 37 84 38 85 33 (1F)
:08C1 84 34 A5 2D A4 2E 85 2F (BB)
:08C9 84 30 85 31 84 32 4C 1D (C4)
:08D1 A8 E6 7A D0 02 E6 7B A0 (C6)
:08D9 00 B1 7A 60 98 48 8A 48 (DF)
:08E1 E6 7A D0 02 E6 7B A0 00 (FB)
:08E9 B1 7A C9 5F F0 0B 68 AA (87)
:08F1 68 A8 60 A2 FB 9A 4C 74 (90)
:08F9 A4 20 20 9B C9 24 F0 0B (5E)
:0901 C9 43 F0 0D C9 44 D0 E6 (C2)
:0909 4C 0D 9F 20 66 9B 4C 42 (99)
:0911 9B 20 89 9A 4C 42 9B A9 (85)
:0919 0E A0 9C 20 1E AB AE F0 (9E)
:0921 9F D0 0A A9 C2 A0 9C 20 (F9)
:0929 1E AB 4C E9 9B A9 00 85 (53)
:0931 02 A5 02 AA E8 A9 00 20 (B2)
:0939 CD BD A9 03 85 D3 A9 22 (CA)
:0941 20 D2 FF A5 02 0A 0A 0A (7B)
:0949 0A AA A0 10 78 E6 01 BD (7B)
:0951 13 A0 F0 0C C6 01 20 D2 (01)
:0959 FF E8 88 D0 EF 78 E6 01 (D6)
:0961 A6 02 BD 03 A0 38 FD 02 (C2)
:0969 A0 85 FE BD F2 9F 38 FD (EE)
:0971 F1 9F 85 FD B0 02 C6 FE (02)
:0979 C6 01 A9 22 20 D2 FF A9 (9A)
:0981 16 85 D3 A5 FE A6 FD 20 (7C)
:0989 CD BD A9 59 A0 9C 20 1E (D0)
:0991 AB E6 02 A5 02 CD F0 9F (0B)
:0999 90 97 A9 0D 20 D2 FF 78 (D4)
:09A1 E6 01 AC F0 9F B9 F1 9F (56)
:09A9 49 FF AA B9 02 A0 49 FF (9C)
:09B1 C6 01 20 CD BD A9 4B A0 (CA)
:09B9 9C 4C 1E AB 00 00 00 0D (64)
:09C1 05 52 41 4D 2D 44 49 53 (7A)
:09C9 4B 2D 36 34 20 56 31 20 (E4)
:09D1 28 57 29 42 59 20 50 4F (58)
:09D9 4C 4F 4E 55 53 0D 28 43 (27)
:09E1 29 31 39 39 33 20 43 4F (10)
:09E9 4D 4D 4F 44 4F 52 45 20 (30)
:09F1 4B 45 42 41 42 20 4C 54 (57)
:09F9 44 0D 0D 00 20 46 52 45 (31)
:0A01 45 20 52 41 4D 2D 44 49 (3D)
:0A09 53 4B 20 42 59 54 45 53 (94)
:0A11 0D 00 0D 4E 49 45 20 50 (F2)
:0A19 4F 44 41 4E 4F 20 4E 41 (6A)
:0A21 5A 57 59 21 0D 00 0D 4A (AE)
:0A29 45 53 54 20 31 36 20 5A (83)
:0A31 42 49 4F 52 4F 57 21 0D (28)
:0A39 00 0D 4E 41 5A 57 41 20 (DE)
:0A41 4A 55 5A 20 49 53 54 4E (E8)
:0A49 49 45 4A 45 21 0D 00 0D (73)
:0A51 5A 41 20 4D 41 4C 4F 20 (01)
:0A59 50 41 4D 49 45 43 49 21 (32)
:0A61 0D 00 0D 44 41 4E 45 20 (AB)
:0A69 5A 41 50 49 53 41 4E 45 (D2)
:0A71 2E 0D 00 42 52 41 4B 20 (F8)
:0A79 5A 42 49 4F 52 4F 57 2E (BD)
:0A81 0D 00 0D 4E 41 5A 57 41 (C1)
:0A89 20 4E 49 45 20 5A 4E 41 (24)
:0A91 4C 45 5A 49 4F 4E 41 21 (D1)
:0A99 0D 00 0D 44 41 4E 45 20 (E3)
:0AA1 4F 44 43 5A 59 54 41 4E (9F)
:0AA9 45 2E 0D 00 0D 5A 42 49 (EE)
:0AB1 4F 52 20 53 4B 41 53 4F (14)
:0AB9 57 41 4E 59 2E 0D 00 A5 (46)
:0AC1 BA C9 0C F0 03 4C ED F5 (F5)
:0AC9 A5 B7 D0 09 A9 61 A0 9C (4D)
:0AD1 20 1E AB 18 60 AC F0 9F (08)
:0AD9 C0 10 90 06 A9 75 A0 9C (D6)
:0AE1 D0 EE 78 E6 01 20 17 9E (ED)
:0AE9 90 08 C6 01 A9 88 A0 9C (A6)
:0AF1 D0 DE A0 00 B1 AE 8D 0D (33)
:0AF9 9C 88 C6 AF B1 AE AA E8 (2C)
:0B01 8A E6 AF C8 91 AE AE F0 (BA)
:0B09 9F BD F1 9F BC 02 A0 85 (BC)
:0B11 FB 84 FC AE F0 9F 8A 0A (4B)
:0B19 0A 0A 0A AA A0 00 B1 BB (D7)
:0B21 9D 13 A0 E8 C8 C4 B7 90 (70)
:0B29 F5 C0 10 B0 09 A9 00 9D (A4)
:0B31 13 A0 E8 C8 D0 F3 A5 C1 (B4)
:0B39 20 FF 9D A5 C2 20 FF 9D (38)
:0B41 A0 00 B1 C1 C8 F0 04 D1 (2F)
:0B49 C1 F0 F9 8D 0B 9C 8C 0C (27)
:0B51 9C C9 B7 F0 0D C0 03 B0 (C5)
:0B59 09 20 FF 9D 20 CC 9D 4C (31)
:0B61 8F 9D A9 B7 20 FF 9D AD (59)
:0B69 0B 9C 20 FF 9D AD 0C 9C (66)
:0B71 20 FF 9D AA 20 CC 9D CA (1C)
:0B79 D0 FA 4C 8F 9D E6 C1 D0 (A4)
:0B81 02 E6 C2 38 A5 C1 E5 AE (F2)
:0B89 A5 C2 E5 AF B0 01 60 68 (7E)
:0B91 68 C6 01 AD 0D 9C 91 AE (97)
:0B99 AE F0 9F E8 8E F0 9F A5 (96)
:0BA1 FB 9D F1 9F A5 FC 9D 02 (AC)
:0BA9 A0 A9 B1 A0 9C 4C 1F 9D (CE)
:0BB1 A0 00 91 FB E6 FB D0 04 (2B)
:0BB9 E6 FC F0 01 60 68 68 C6 (CE)
:0BC1 01 A9 9E A0 9C 4C 1F 9D (0E)
:0BC9 AD F0 9F D0 02 18 60 A5 (E0)
:0BD1 B7 C9 11 90 02 A9 10 85 (30)
:0BD9 B7 A9 00 85 02 A0 00 A5 (F3)
:0BE1 02 0A 0A 0A 0A AA BD 13 (39)
:0BE9 A0 D1 BB D0 0D E8 C8 C4 (F0)
:0BF1 B7 90 F3 BD 13 A0 D0 02 (7F)
:0BF9 38 60 E6 02 A5 02 CD F0 (16)
:0C01 9F 90 DA 18 60 48 A5 BA (9D)
:0C09 C9 0C F0 04 68 4C A5 F4 (C9)
:0C11 68 A5 B7 D0 03 4C 1B 9D (B0)
:0C19 78 E6 01 20 17 9E 08 C6 (7B)
:0C21 01 28 B0 07 A9 D1 A0 9C (1D)
:0C29 4C 1F 9D A6 02 78 E6 01 (5A)
:0C31 BD F1 9F 85 AC BD 02 A0 (A5)
:0C39 85 AD BD F2 9F 85 AE BD (06)
:0C41 03 A0 85 AF A0 00 A5 B9 (46)
:0C49 F0 0D 20 DC 9E 85 C2 20 (B1)
:0C51 DC 9E 85 C3 4C B4 9E 20 (16)
:0C59 DC 9E 20 DC 9E A5 2B 85 (96)
:0C61 C2 A5 2C 85 C3 20 DC 9E (94)
:0C69 C9 B7 F0 06 20 D3 9E 4C (D8)
:0C71 B4 9E 20 DC 9E 48 20 DC (C3)
:0C79 9E AA 68 20 D3 9E CA D0 (08)
:0C81 FA 4C B4 9E 91 C2 E6 C2 (6E)
:0C89 D0 02 E6 C3 60 38 A5 AC (3A)
:0C91 E5 AE A5 AD E5 AF B0 09 (2C)
:0C99 B1 AC E6 AC D0 02 E6 AD (DE)
:0CA1 60 68 68 C6 01 A9 E9 A0 (87)
:0CA9 9C 20 1E AB A6 C2 A4 C3 (F5)
:0CB1 18 60 A9 FB A0 9C C6 01 (B6)
:0CB9 20 1E AB 4C 42 9B 20 20 (1E)
:0CC1 9B C9 22 F0 03 4C 42 9B (9D)
:0CC9 AD F0 9F F0 F8 A6 7A A4 (31)
:0CD1 7B E8 D0 01 C8 86 BB 84 (E5)
:0CD9 BC A2 00 20 20 9B C9 22 (36)
:0CE1 F0 05 E8 E0 11 90 F4 86 (B0)
:0CE9 B7 E0 00 F0 D8 78 E6 01 (86)
:0CF1 20 17 9E 08 C6 01 28 B0 (C1)
:0CF9 0A A9 D1 A0 9C 20 1E AB (4A)
:0D01 4C 42 9B 78 E6 01 A5 02 (A6)
:0D09 AA E8 EC F0 9F D0 06 CE (A9)
:0D11 F0 9F 4C 01 9F 0A 0A 0A (21)
:0D19 0A AA 18 69 10 A8 B9 13 (57)
:0D21 A0 9D 13 A0 E8 C8 D0 F6 (59)
:0D29 A6 02 BD F1 9F 85 FB BD (D9)
:0D31 02 A0 85 FC BD 03 A0 38 (E2)
:0D39 FD 02 A0 85 FE BD F2 9F (35)
:0D41 38 FD F1 9F 85 FD B0 02 (36)
:0D49 C6 FE BD F2 9F 9D F1 9F (6F)
:0D51 BD 03 A0 9D 02 A0 E8 E0 (97)
:0D59 11 90 EF A6 02 BD F1 9F (03)
:0D61 85 20 BD 02 A0 85 21 A0 (97)
:0D69 00 B1 20 91 FB E6 FB D0 (24)
:0D71 02 E6 FC E6 20 D0 04 E6 (A4)
:0D79 21 F0 05 4C B8 9F A6 02 (B2)
:0D81 BD 02 A0 38 E5 FE 9D 02 (D7)
:0D89 A0 BD F1 9F E5 FD 9D F1 (39)
:0D91 9F B0 03 DE 02 A0 E8 E0 (40)
:0D99 11 90 E5 CE F0 9F 4C 01 (44)
:0DA1 9F 00 A0 27 B9 00 02 99 (3C)
```

Dylematy Dr. Boczka...

Cześć Kebabie !

Chciałbym pomóc w rozwiązaniu problemu z 10 numeru Kebaba, gdzie Dr.Boczek ma ogromny problem ze swą Amigą. Stwierdził on bowiem, że uruchamia mu się Deluxe Paint, a posiada tylko pamięć Chip. Ja też już miałem kiedyś taki problem i nie mogłem stwiedzić co było tego przyczyną. Udałem się ze swą nowo zakupioną Amigą do serwisu RTV. Po rozebraniu sprzętu nic nie stwierdzono. Fachowiec stwierdził, że kości są na miejscu a śladów wirusa nie widać. Być może był to wirus, który działał we wnętrzu układów.

Przez około miesiąc czasu chodziłem od serwisu do serwisu i trafiłem na prawdziwego fachowca, który stwierdził, że układ Fat Agnus nadaje się tylko do wymiany gdyż jest zarażony specjalnym wirusem plikowym!!! Na pewno Amiga Dr.Boczka jest zarażona wyżej wspomnianym wirusem plikowym albo układ Fat Agnus jest uszkodzony skoro program Deluxe Paint uruchamia się tylko w pamięci Chip to w tym musi tkwić przyczyna.

Porady amatorskiemu pismu Kebab, udzielił

ENIACS

Dziękujemy niezmiernie, drogi kolego za pomoc w imieniu Dr. Boczka, niestety jak wynika z kolejnych listów, które od niego otrzymaliśmy, zaistniały nowe problemy. Być może i tym razem będziesz mógł pomóc? Jeżeli tak, to prosimy bardzo o ponowny kontakt.

Drogi KEBAB'ie,

zawsze chciałem zostać grafikiem. W jednym z numerów profesjonalnego miesięcznika fanów magazynu AMIGA, przeczytałem, że najlepszym komputerem dla grafików jest Amiga 3000T ze względu na posiadane 4MB Chip-RAM'u. Idąc za radą profesjonalistów (i profesjonalnym, kolorowym zdjęciem) oddałem koledze za pół darmo moją A500 (bez Fast-RAM'u) i nabyłem A3000T.

Ku mojemu zdumieniu po podłączeniu nowego nabytku, Workbench 2.0 nie miał najmniejszego zamiaru pokazać oczekiwanych czterech megabajt`w "graphics mem". Czy ja muszę mieć takiego pecha? Jak ja teraz zostanę grafikiem? Wszyscy koledzy się ze mnie śmieją! Powiedzcie co robić !? Czy to możliwe, żeby wirus plikowy przeskoczył z mojej starej A500 do A3000 kiedy stały przez kilka minut jedna na drugiej (tworząc plik) ??!

Dr. Boczek

Jest nam niezmiernie przykro, ale niestety i tym razem nie jesteśmy w stanie Ci pomóc drogi Doktorze Boczku. Twoje problemy należą do gatunku w pełni profesjonalnych a to nie jest niestety nasza dziedzina. Gdyby przytrafiły Ci się jakieś bardziej amatorskie kłopoty. to istnieje duże prawdopodobieństwo, że potrafilibyśmy je rozwiązać. Tymczasem musimy ponownie odwołać się do wiedzy i doświadczenia naszych Drogich Czytelników. Może ktoś z nich (np. ENIACS) potrafiłby Ci pomóc. Czekamy na listy!

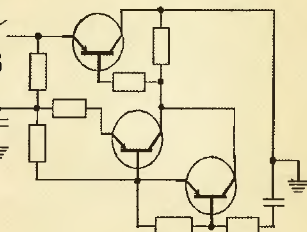
Drogi KEBAB'ie,

w jednym z numerów profesjonalnego magazynu fanów miesięcznika AMIGA, przeczytałem, że aby zamknąć okienko Workbench'a należy dwukrotnie kliknąć (dwumysznąć) na "gadżecie" zamykania okna. Dopóki miałem do czynienia z profesjonalnym systemem operacyjnym "Windows" na profesjonalnych komputerach typu iBM pc, to wszystko działało jak trzeba (czasami udawało mi się kliknąć nawet pięć razy). Niestety moja zabawka (A3000T) nie pozwala mi na zastosowanie się do zaleceń profesjonalistów! Okienko znika zanim zdążę pomyśleć o drugim kliknięciu!!! Poczułem się w tym momencie totalnym amatorem, więc postanowiłem napisać do was. Może znacie jakieś ćwiczenia dla palców, które pozwolą mi nabrać wystarczającej szybkości w klikaniu?

Dr. Boczek

Nareszczie coś w czym chyba możemy Ci pomóc! Jednym sposobem na rozwiązanie tego problemu jest zapisanie się na prywatne lekcje do jakiegoś znanego pianisty np. p. Zimmermann'a. Powinno to wydatnie podnieść sprawność (i szybkość) Twoich palców. Ale istnieje też szybsza (i tańsza) metoda na ten problem. Wystarczy zamontować do Twojej myszki AUTO-FIRE! Schemat takiego układu zamieszczamy poniżej. Przy odpowiednim dobraniu elementów powinieneś ponownie osiągnąć profesjonalny poziom pięciu kliknięć na jedno okienko (a właściwie "gadżet")

Twój "KEBAB"



Kupon ogłoszeniowy

imię i nazwisko

adres

treść:

